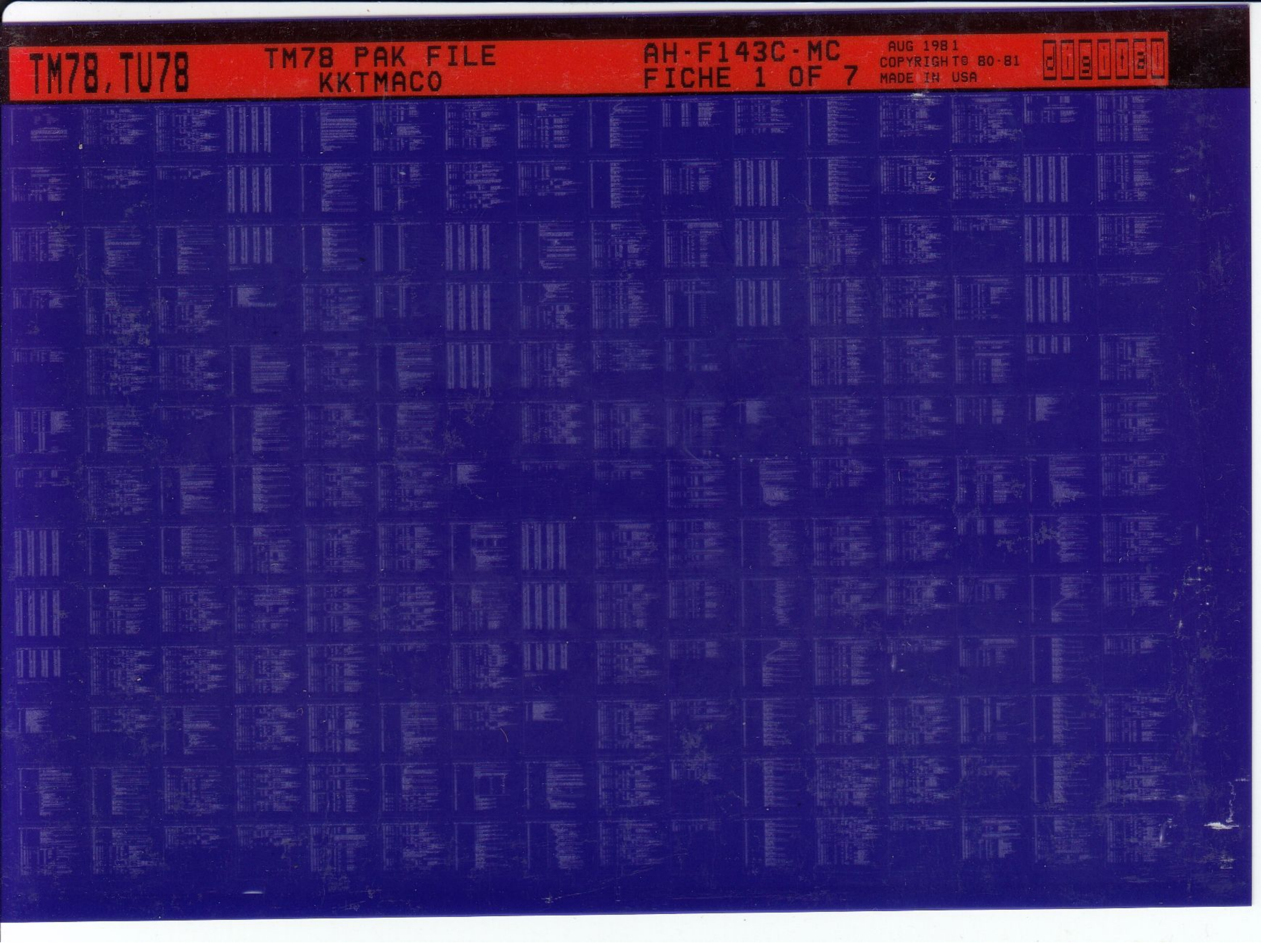
TM78,TU78

TM78 PAK FILE
KKTMACO

AH-F143C-MC
FICHE 1 OF 7



TM78,TU78 TM78 PAK FILE KKTMACO AH-F143C-MC FICHE 2 OF 7

TM78,TU78 TM78 PAK FILE AH-F143C-MC AUG 1981
KKTMACO FICHE 3 OF 7

TM78,TU78 TM78 PAK FILE KKTMACO AH-F143C-MC FICHE 4 OF 7

digital

TM78,TU78 TM78 PAK FILE KKTMACO AH-F143C-MC FICHE 5 OF 7

TM78,TU78 TM78 PAK FILE KKTMACO AH-F143C-MC FICHE 6 OF 7 AUG 1981 COPYRIGHT© 80-81 MADE IN USA

TM78,TU78 TM78 PAK FILE KKTMACO AH-F143C-MC FICHE 7 OF 7 AUG 1981

```
1329                            .REM %
1330
1331
1332
1333
1334                                        IDENTIFICATION
1335                                        --------------
1336
1337                                PRODUCT CODE :          AC-F144C-MC
1338
1339                                PRODUCT NAME :          KKTMAC0 TM78 PAK FILE
1340
1341                                MAINTAINER :            DIAGNOSTIC ENGINEERING
1342
1343                                DATE :                  FEBRUARY 1, 1981
1344
1345                                AUTHOR :                G. COOKE
1346
1347
1348
1349
1350                                        COPYRIGHT (C) 1980,1981 BY
1351                                DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
1352
1353                    THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
1354                    ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
1355                    INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
1356                    COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
1357                    OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
1358                    TRANSFERRED.
1359
1360                    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
1361                    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
1362                    CORPORATION.
1363
1364                    DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
1365                    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
1366
1367
1368
1369
1370                            %
```

```
                                        .TITLE  BCT1 - BASIC CONFIDENCE TESTS
                                        .SBTTL  CPU/CAS INTERFACE PROGRAM
                                        ;ID     BCT1-BASIC CONFIDENCE TEST

4300                                    S
(1)                                     ; ************************************************************************
                                        ;THE VALUE IN THE RAM LOCATION CMD WILL CONTROL THE FUNCTION OF THE
                                        ;PROGRAM AS PER THE FOLLOWING TABLE.
                                        ;
                                        ;               CMD                     FUNCTION
                                        ;
                                        ;                0              READ THE CAS REGISTERS
                                        ;                1              WRITE THE CAS REGISTERS
                                        ;                2              CAS CONTENTION TEST
                                        ;                3              RAM MEMORY TEST
                                        ;
4300                                    S
(1)                                     ; ************************************************************************

                4300                    .=      DIAGPG
4300  C3  11  43       10.0  START:    JMP     PROCESS         ;THEN EXECUTE TEST

4303                                    TESTX   @1              ;DUMMY INITIALIZE MACRO
(1) 4303  3E  01           7.0                          MVI     A,@1            ;DEFINE THE TEST NUMBER
(1) 4305  CD  03  28      18.0                          CALL    TSET            ;SETUP THE TEST
                                        ;%BASIC CONFIDENCE TEST DUMMY TEST HEADER
                                        ;&DUMMY MODULE CALLOUT HEADER
                4310                    .=      $4310           ;ADDRESS OF THE ''CMD'' WORD
4310  00                        CMD:    .BYTE   0               ;ASSUME READ CYCLE

4311  3A  10  43       13.0  PROCESS: LDA     CMD             ;GET ADDR OF COMMAND CONTROL WORD
4314  A7            4.0              ANA     A               ;RESET CONDITION BITS
4315  CA  5B  43      10.0              JZ      READ            ;IF =0 READ, IF =1 LOAD CAS
4318  FE  01           7.0              CPI     $01             ;LOAD COMMAND?
431A  CA  25  43      10.0              JZ      LOAD            ;YES-GO LOAD CAS REGISTERS
431D  FE  02           7.0              CPI     $02             ;CAS CONTENTION TEST?
431F  CA  73  43      10.0              JZ      CASCTN          ;YES - GO DO THE CONTENTION THING
4322  C3  8C  43      10.0              JMP     MEM             ;NO - ASSUME MEMORY TEST
```

```
                                        .SBTTL  WRITE THE CAS REGISTERS ROUTINE

                                LOAD:   LXI     H,R01L.MM       ;GET 1ST CAS ADDRESS
                                        LDA     LDAT            ;GET LOW BYTE TO LOAD.
                                        MOV     B,A             ;PUT DATA IN REG B

                                LOADL:  NOP                     ;HACK-TO ALLOW PROPER RETURN FROM CONTENTION INTERRUPT T
                                        MOV     M,B             ;PUT CONTENTS OF ''CMD'' IN ALL CAS REGISTERS
                                        MOV     A,A             ;HACK
                                        INX     H               ;UPDATE THE CAS POINTER
                                        INX     H
                                        MOV     A,L             ;GET LOW BYTE OF POINTER
                                        CPI     R17L+2          ;DONE? (1ST ADDRESS OF HIGH BYTE AREA)
                                        JC      LOADL           ;NO - DO NEXT CAS ADDRESS

                                LOADH:  LXI     H,R01H!MM
                                        LDA     HDAT            ;GET HIGH BYTE DATA WORD
                                        MOV     B,A             ;SAVE THE DATA BYTE IN REG B
                                LDHILP: NOP                     ;HACK-TO ALLOW PROPER RETURN FORM CONTENTION INTERRUPT T
                                        MOV     M,B             ;STORE IN HIGH BYTE ADDRESS
                                        MOV     A,A             ;HACK
                                        INX     H               ;POINT TO NEXT ADDRESS
                                        INX     H
                                        MOV     A,L             ;GET LOW BYTE OF ADDRESS POINTER
                                        CPI     R17H+2          ;DONE WITH HIGH BYTE ADDRESSES?
                                        JC      LDHILP          ;NO - LOAD NEXT HIGH ADDRESS
                                DONE:   MVI     A,@372          ;GET TERMINATE #
                                        ROUT    R00L            ;TELL OUTSIDE WORLD
                                (1)     OUT     R00L                    ;WRITE AC INTO R00L
                                (1)     MOV     A,A                     ;RETRY LINK
                                DONEX:  RIN     R00L            ;GET THE FUNCTION CODE REGISTER
                                (1)     IN      R00L                    ;READ R00L INTO AC
                                (1)     MOV     A,A                     ;RETRY LINK
                                        ANI     @77             ;DID THE HOST ISSUE
                                        CPI     @35             ;RESTART?
                                        JNZ     DONEX           ;NO - KEEP ON LOOKING
                                        JMP     PROCESS         ;YES - DO IT AGAIN
```

```
                                        .SBTTL  READ THE CAS REGISTERS ROUTINE
                                        ;HERE TO READ THE CAS REGISTERS FROM THE MICROCOMPUTER SIDE AND SAVE
                                        ;THE DATA BYTE READ IN "CSAV" STORAGE AREA

1445  435B  21  82  80      10.0  READ:   LXI     H,R01L!MM       ;SETUP CAS POINTER
1446  435E  11  00  44      10.0          LXI     D,CSAV          ;AND POINTER TO STORAGE
1447
1448  4361  00               4.0  READL:  NOP                     ;HACK-TO ALLOW PROPER RETURN FROM CONTENTION INTERRUPT T
1449  4362                           RMOVAM
 (1)  4362  00               4.0                  NOP                     ;FOR INTERRUPT SERVICE SEARCH
 (1)  4363  7E               7.0                  MOV     A,M             ;READ MEMORY TO AC
 (1)  4364  7F               4.0                  MOV     A,A             ;RETRY LINK
1450  4365  EB               4.0          XCHG                    ;SWAP D&E WITH H&L
1451  4366  77               7.0          MOV     M,A             ;SAVE THE CAS CONTENTS
1452  4367  EB               4.0          XCHG                    ;RESET D&E AND H&L
1453  4368  23               6.0          INX     H               ;UPDATE CAS POINTER
1454  4369  13               6.0          INX     D               ;AND THE STORAGL POINTER
1455  436A  7D               4.0          MOV     A,L             ;GET LOW BYTE OF CAS POINTER
1456  436B  FE  A0            7.0          CPI     R17H+1          ;DONE YET?
1457  436D  DA  61  43       10.0          JC      READL           ;NO - DO NEXT CAS ADDRESS
1458  4370  C3  49  43       10.0          JMP     DONE            ;WAIT FOR A RESTART COMMAND
1459
```

```
                                                .SBTTL  CAS CONTENTION INT. ROUTINE

                                        4.0  CASCTN: XRA     A               ;CLEAR THE CAS CONTENTION
4373  AF                                13.0         STA     CASCT           ;INTERRUPT
4374  32    21    4F                    10.0         LXI     H,ROOL!MM               ;LOAD STARTING ADDRESS OF CAS
4377  21    80    80                     7.0  CASLP1: MOV     A,M             ;READ CAS
437A  7E                                 6.0         INX     H               ;UPDATE THE COUNTER
437B  23                                 4.0         MOV     A,L             ;CHECK FOR DONE
437C  7D                                 7.0         CPI     @241            ;
437D  FE    A1                          10.0         JNZ     CASLP1          ;CONTINUE UNTIL ALL READ
437F  C2    7A    43                    13.0         LDA     CASCT           ;GET CAS CONTENTION COUNT
4382  3A    21    4F                     4.0         ANA     A
4385  A7                                10.0         JZ      CASCTN          ;NO ERROR-CONTINUE
4386  CA    73    43                    10.0         JMP     DONE            ;WAIT FOR A RESTART COMMAND
4389  C3    49    43
```

```
1476                                        ;MICROCOMPUTER MEMORY EXERCISER ROUTINE
1477
1478  438C  21  B9  43      10.0  MEM:   LXI   H,MDAT          ;POINT TO MEMORY DATA PATTERN TABLE
1479  438F  46                7.0  MEM1:  MOV   B,M            ;LOAD REG B WITH DATA PATTERN
1480  4390  EB                4.0         XCHG                 ;PUT H&L PATTERN POINTER IN D&E
1481  4391  21  22  44      10.0         LXI   H,ENDPGM       ;POINT TO 1ST FREE ADDRESS TO TEST
1482  4394  70                7.0  MEM2:  MOV   M,B            ;PUT DATA IN REGISTER "A" IN MEMORY
1483  4395  7E                7.0         MOV   A,M            ;GET IT FROM MEMORY
1484  4396  B8                4.0         CMP   B              ;READ/VERIFY DATA IN MEMORY
1485  4397  C2  AC  43      10.0         JNZ   MEMHLT         ;DIE IF ERROR DETECTED
1486  439A  23                6.0         INX   H              ;UPDATE THE MEMORY POINTER
1487  439B  7C                4.0         MOV   A,H            ;GET ADDR HI BYTE
1488  439C  FE  50             7.0         CPI   $50            ;DONE WITH MEMORY (16K)??
1489  439E  DA  94  43      10.0         JC    MEM2           ;LOOP AGAIN IF NOT DONE WITH MEMORY PASS
1490  43A1  A7                4.0  MEMDAT: ANA   A              ;CLEAR THE CONDITION BITS
1491  43A2  78                4.0         MOV   A,B            ;PUT CURRENT PATTERN IN REG A
1492  43A3  3C                4.0         INR   A              ;SEE IF REG A GOES TO 0
1493  43A4  CA  49  43      10.0         JZ    DONE           ;PUT PATTERN POINTER IN H&L REG
1494  43A7  13                6.0         INX   D              ;POINT TO NEXT PATTERN TO USE FROM TABLE
1495  43A8  EB                4.0         XCHG                 ;PUT THE PATTERN POINTER IN H & L REGS
1496  43A9  C3  8F  43      10.0         JMP   MEM1           ;TEST NEXT MEM ADDRESS
1497  43AC  22  CD  43      16.0  MEMHLT: SHLD  BADR           ;STORE THE BAD ADDRESS INFO
1498  43AF  32  CF  43      13.0         STA   BDAT           ;SAVE THE DATA BYTE IN USE WHEN ERROR DETECTED
1499  43B2  78                4.0         MOV   A,B            ;PUT REAL DATA IN A
1500  43B3  32  D0  43      13.0         STA   ADAT           ;SAVE IT ALSO
1501  43B6  C3  B6  43      10.0         JMP   .              ;STAY HERE FOREVER...OR AT LEAST A WHILE
1502                                    ;HERE IS THE TABLE OF DATA PATTERNS TO USE THROUGH MEMORY
1503  43B9  00                        MDAT:  .BYTE  @000          ;ALL 0'S
1504  43BA  01                               .BYTE  @001          ;FLOATING 1'S
1505  43BB  02                               .BYTE  @002
1506  43BC  04                               .BYTE  @004
1507  43BD  08                               .BYTE  @010
1508  43BE  10                               .BYTE  @020
1509  43BF  20                               .BYTE  @040
1510  43C0  40                               .BYTE  @100
1511  43C1  80                               .BYTE  @200
1512  43C2  7F                               .BYTE  @177          ;FLOATING 0'S
1513  43C3  BF                               .BYTE  @277
1514  43C4  DF                               .BYTE  @337
1515  43C5  EF                               .BYTE  @357
1516  43C6  F7                               .BYTE  @367
1517  43C7  FB                               .BYTE  @373
1518  43C8  FD                               .BYTE  @375
1519  43C9  FE                               .BYTE  @376
1520  43CA  AA                               .BYTE  @252          ;ALTERNATE BITS
1521  43CB  55                               .BYTE  @125
1522  43CC  FF                               .BYTE  @377          ;LAST PATTERN
1523                                  .SBTTL  PROGRAM VARIABLES
1524  43CD  00  00                    BADR:  .WORD  0             ;ADDRESS OF THE FAILING ADDRESS
1525  43CF  00                        BDAT:  .BYTE  0             ;FAILING DATA
1526  43D0  00                        ADAT:  .BYTE  0             ;EXPECTED DATA
```

```
1528          4400                        .=      $4400           ;START OF CAS REGISTER STORAGE AREA
1529  4400     00                  CSAV:   .BYTE   0               ;REG 00 LOW
1530  4401     00                          .BYTE   0               ;REG 00 HIGH
1531  4402     00                          .BYTE   0               ;REG 01 LOW
1532  4403     00                          .BYTE   0               ;REG 01 HIGH
1533  4404     00                          .BYTE   0               ;REG 02 LOW
1534  4405     00                          .BYTE   0               ;REG 02 HIGH
1535  4406     00                          .BYTE   0               ;REG 03 LOW
1536  4407     00                          .BYTE   0               ;REG 03 HIGH
1537                                               ;ETC.
1538                                               ;ETC.
1539
1540          4420                        .=      $4420
1541
1542                              ;''LDAT'' FOR LOW BYTE OF CAS REGISTERS 0 - 17
1543                              ;''HDAT'' FOR HIGH BYTE OF CAS REGISTERS 0 - 17
1544
1545  4420     00                  LDAT:   .BYTE   0               ;STORAGE FOR LOW BYTE DATA
1546  4421     00                  HDAT:   .BYTE   0               ;STORAGE FOR HIGH BYTE DATA
1547  4422 0000                    ENDPGM: .END
```

```
A       =%0007  ADAT    43D0  ADATA  = 0094  AMTIEP= 0001
AMTIE7= 0002  ARAIDF= 0098  ASAVE   4F9B  ATTCD   4F97
AXNUM   4F91  B       =%0000  BADR    43CD  BADST  = 0090
BDAT    43CF  BIT0    = 0001  BIT1   = 0002  BIT15  = 8000
BIT2    = 0004  BIT3    = 0008  BIT4   = 0010  BIT5   = 0020
BIT6    = 0040  BIT7    = 0080  BIT8   = 0100  BIT9   = 0200
BRKPBC= 4F0A  BRKRAM= 4F10  BRKSTR= 4E60  BRKXCT= 4F00
BSAVE   4F9C  BYTCNT= 00D4  BYTEH   4F24  BYTEL   4F23
C       =%0001  CASCT   4F21  CASCTL= 00A0  CASCTN  4373
CASLP1  437A  CASSTA= 00A0  CATTH  = 0089  CATTL  = 0088
CBUSST= 00A1  CBYTH   = 008B  CBYTL  = 008A  CDG1H  = 0087
CDG1L   = 0086  CDG2H   = 0093  CDG2L  = 0092  CDG3H  = 0095
CDG3L   = 0094  CDVTH   = 008D  CDVTL  = 008C  CHPTIE= 0028
CH0TIE= 0020  CH1TIE= 0021  CH2TIE= 0022  CH3TIE= 0023
CH4TIE= 0024  CH5TIE= 0025  CH6TIE= 0026  CH7TIE= 0027
CKLOP   = 2815  CLKCTL= 00F0  CLOCK   4F26  CMC0H  = 0099
CMC0L   = 0098  CMC1H   = 009B  CMC1L  = 009A  CMC2H  = 009D
CMC2L   = 009C  CMC3H   = 009F  CMC3L  = 009E  CMD      4310
CMINH   = 0097  CMINL   = 0096  CNTCTL= 00D7  CRCWRD= 0018
CSAV    4400  CSAVE   4F9D  CSRLH  = 0091  CSRLL  = 0090
CTCH    = 0085  CTCL    = 0084  CTSTH  = 008F  CTSTL  = 008E
CXCTH   = 0081  CXCTL   = 0080  CXINH  = 0083  CXINL  = 0082
C.      = 0001  C.AVAI= 0080  C.DP   = 0008  C.DSE  = 0010
C.DTU   = 0003  C.DVA   = 0008  C.FAIL= 00FC  C.FMT  = 0070
C.FNCT= 003E  C.GO    = 0001  C.INTC= 00FE  C.MAIN= 0020
C.NSA   = 0080  C.RCT   = 00FC  C.SER  = 0080  C.SHR  = 0040
C.SKPC= 000F  C.TAPE= 0040  C.WCS  = 0002  D       =%0002
DATACT= 00D0  DBUS    4F28  DBUSCT= 00C0  DBUSST= 00C0
DDRA    = 00D8  DDRAIN= 0010  DDRB   = 00D9  DDRBIN= 0002
DDRC    = 00DA  DDRCIN= 0001  DDRCO  = 0088  DDRCTL= 00DB
DIAFLG  4F22  DIAGPG= 4300  DIAGRM= 4F90  DIARD  = 0008
DONE    4349  DONEX   434E  DONE1  = 0045  DONINT= 0010
DSAVE   4F9E  DSE     = 0006  D.ATH0= 0001  D.ATH1= 0002
D.EOTD= 0010  D.LAGC= 0020  D.NOTW= 0040  D.NTHR= 0004
D.TACH= 0008  D.WR4   = 0080  E       =%0003  ECCBAD= 0042
ECCCOR= 0019  ECCOK   = 0041  ECCSTA= 001A  ECCTST= 000E
EDATA   = 0095  ENDPGM  4422  EOTCLR= 0003  ERFLG   4F93
ERLP    = 2809  ERLPA   = 280F  ERLPB  = 2812  ERLPE  = 280C
ERNUM   4F90  ERRCNT= 00D6  ESAVE   4F9F  E.ACRC= 0010
E.AMT   = 0020  E.CDP   = 0080  E.CRC  = 0080  E.PNTR= 0008
E.RPE   = 0040  E.STEC= 0001  E.TTEC= 0002  E.UNC  = 0004
FIFORD= 006A  FORMAT  4F25  FWDTST= 0061  GCRID  = 0089
GCRSET= 0002  GOODTM= 0092  H       =%0004  HDAT     4421
HLSAVE  4FA0  IE      = 0008  INTSTA= 00E0  ITERA   4F9A
I.PWR   = 0020  I.RMPE= 0040  I5.5   = 0010  I6.5   = 0020
I7.5    = 0040  KCALL   = 005F  KCLR   = 007B  KDEP   = 003F
KENAB   = 0078  KEXAM   = 003E  KEYBRD= 00C8  KEY1   = 0078
KEY10   = 006D  KEY11   = 006E  KEY12  = 006F  KEY13  = 005C
KEY14   = 005D  KEY15   = 005E  KEY16  = 005F  KEY17  = 003C
KEY18   = 003D  KEY19   = 003E  KEY2   = 0079  KEY20  = 003F
KEY3    = 007A  KEY4    = 007B  KEY5   = 0074  KEY6   = 0075
KEY7    = 0076  KEY8    = 0077  KEY9   = 006C  KINTA  = 006F
KLDAD   = 003D  KN0     = 003C  KN1    = 005C  KN2    = 005D
KN3     = 005E  KN4     = 006C  KN5    = 006D  KN6    = 006E
```

```
KN7    = 0074          KN8    = 0075          KN9    = 0076          KU2    = 0079
KU3    = 007A          KU8    = 0077          L      =%0005          LBLANK= 000F
LCE    = 0008          LCH    = 000C          LCL    = 000D          LCP    = 000E
LC0    = 0000          LC1    = 0001          LC2    = 0002          LC3    = 0003
LC4    = 0004          LC5    = 0005          LC6    = 0006          LC7    = 0007
LC8    = 0008          LC9    = 0009          LDAT     4420          LDHILP  433E
LDLEDA= 00CA           LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046          LOAD     4325
LOADH    4337          LOADL    432C          LPFLG    4F94          LPNUM    4F92
M      =%0006          MBSEL  = 00E0          MB.A   = 0008          MB.B   = 0004
MDAT     43B9          MEM      438C          MEMDAT   43A1          MEMHLT   43AC
MEMTOP= 4FFF           MEM1     438F          MEM2     4394          MINUS  = 000A
MM     = 8000          MSE    = 0008          MSGN   = 0000          MTACLR= 0000
MT.ARA= 0020           MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040
MT.INH= 0008           MT.LWR= 0004           MT.MOT= 0002           MT.NWT= 0080
MT.PEC= 0040           MT.PSB= 0004           MT.PS0= 0001           MT.PS1= 0002
MT.REV= 0020           MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080
M.CAPE= C020           M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010
M.EXC  = 0008          M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010
M.OCC  = 0020          M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080
M.RDEN= 0002           M.RDPE= 0008           M.RUN  = 0004          M.SGLK= 0001
M.TRA  = 0040          M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080
M.WREN= 0080           M5.5   = 0001          M6.5   = 0002          M7.5   = 0004
NOTCAP= 0088           OKAY   = 00FF          OPRRAM= 4300           OPSTRT= 0058
OPVER  = 0040          PADCNT= 00D5           PADCRC= 0080           PDIAG  = 0048
PEID   = 008A          PENAB  = 004C          PESET  = 0001          PL     = 00B1
PRDD   = 004C          PRENF  = 009C          PROCES   4311          PS     = 00B2
PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001           P.BCTC= 0040
P.CMDP= 0020           P.INTE= 0080           P.LCS  = 0040          P.LWR  = 0020
P.RDP  = 0002          P.RPEN= 0004           P.RPST= 0002           P.RPOE= 0020
P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010           P.SING= 0080
P.STAT= 0002           P.STPE= 0080           P.TACH= 0008           P.TUPR= 0010
P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001           P.WPEN= 0010
P.WPOE= 0008           P.WP1E= 0004           P.WP2E= 0008           P.WP3E= 0004
P.5VOK= 0002           QUE    = 281B          QUEM   = 281E          RAMT   = 0010
RARA   = 0006          RARAI  = 0004          RCHBD0= 0048           RCHBD1= 0047
RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D          RCMD   = 000B
RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017          RDCLK  = 0010
RDON   = 0011          READ     435B          READG  = 0007          READL    4361
REND   = 0014          REQST  = 2806          RESCHR= C0D1           REVTST= 0064
REWIND= 0004           RFIFOL= 0008           RGCLK  = 0002          RGCRI  = 0003
RIBG   = 0001          RILL   = 0012          RINST  = 000C          RMCTST= 0008
RMK2   = 0013          RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044
RPCHI  = 0001          RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002
RPFAIL= 0000           RPF1   = 009D          RPF2   = 009E          RPOK   = 0043
RPOSTN= 0016           RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002
RTIEB  = 000A          RTIER  = 0030          RTM    = 0005          RUNKI  = 0009
RUPTST= 005E           RWDUNL= 0005           R.AMT  = 0001          R.BOP  = 0008
R.DATA= 0040           R.DON  = 0002          R.DRDY= 0010           R.END  = 0010
R.ILL  = 0004          R.JVOK= 0004           R.MK2  = 0008          R.PLOD= 0008
R.PLO0= 0010           R.PLO1= 0020           R.POST= 0020           R.STNM= 0002
R.STOP= 0004           R.STPC= 0001           R.TBJN= 0080           R.TSTD= 0040
```

```
R.VOK  = 0080    R00H  = 0081    R00L  = 0080    R01H  = 0083
R01L   = 0082    R02H  = 0085    R02L  = 0084    R03H  = 0087
R03L   = 0086    R04H  = 0089    R04L  = 0088    R05H  = 008B
R05L   = 008A    R06H  = 008D    R06L  = 008C    R07H  = 008F
R07L   = 008E    R10H  = 0091    R10L  = 0090    R11H  = 0093
R11L   = 0092    R12H  = 0095    R12L  = 0094    R13H  = 0097
R13L   = 0096    R14H  = 0099    R14L  = 0098    R15H  = 009B
R15L   = 009A    R16H  = 009D    R16L  = 009C    R17H  = 009F
R17L   = 009E    R7.5  = 0010    SELCLR= 0000    SETATA= 00A1
SID    = 0080    SOD   = 0080    SOE   = 0040    SP    =%0008
SSCLK  = 0040    SSTEP = 0005    STACK = 4FFF    START   4300
STATRM= 4F20     STPCT   4F20    STRSP = 5000    TADR00= 0080
TADR01= 0081     TADR02= 0082    TADR03= 0083    TADR04= 0084
TADR05= 0085     TADR06= 0086    TADR07= 0087    TADR10= 0088
TADR11= 0089     TADR12= 008A    TADR13= 008B    TAMT  = 0044
TASEL  = 0080    TCMD  = 0040    TC.FWD= 0040    TC.INH= 0008
TC.LWR= 0004     TC.REV= 0020    TC.WRT= 0010    TEMP    4F99
TMF    = 0099    TMRDY = 0040    TRKENA= 00D2    TSET  = 2803
TSTEND= 2818     TSTS  = 0040    TUSEL0= 00D1    TUSEL1= 00D2
TU78   = C010    T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004
T.EOT  = 0002    T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020
T.PES  = 0008    T.PSBJ= 0020    T.PSOJ= 0008    T.PS1J= 0010
T.RDY  = 0080    T.RDYO= 0040    T.RWD = 0010    T.SCLK= 0002
UIBG   = 00A1    VALFC   4F98    VALTB   4F95    VELTST= 005B
WDR.P  = 0010    WMCCTL= 00D3    WMCERR= 00DA    WMCSTA= 00D0
WRTCLK= 0000     WRTDAT= 00D3    W.ACRC= 0004    W.CRC  = 0008
W.DIAG= 0002     W.DONN= 0040    W.ECC  = 0010   W.ENAB= 0080
W.ERR  = 0020    W.FMT  = 0070   W.GCR  = 0010   W.LEFT= 0004
W.ONES= 0020     W.RESI= 0002    W.REV  = 0004   W.ROME= 0010
W.RST  = 0001    W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020
X      =%000A    X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002
X.ROME= 0001     X.WCLK= 0001    Y      =%000B   .      = 4422


 ERRORS DETECTED:  0

 *,KKTMAC.LST/NL:TOC=NLIST,PARAM,MACRO,LIST,KKTMAC.M80,BCT1
 RUN-TIME: 3 3 0 SECONDS
 CORE USED:  9K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:06 SEQ 0011
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  WMC1 - WRITE MICROCONTROLLER PART #1
                                .SBTTL  TEST 1 - WMC SELF DIAGNOSTICS
                                ;ID     WMC1-WRITE MICRO CONTROLLER PART #1

                                ;THIS SERIES OF WRITE MICROCONTROLLER TESTS ARE DESIGNED TO TEST
                                ;THE WMC INTERFACE LOGIC. THE 'BYTE COUNTER', 'PAD COUNTER', AND
                                ;'ECODE COUNTER' LOGIC IS TESTED AS WELL AS THE GROSS WMC 'READY
                                ;CHECK.
                                ;
                                ;THESE TESTS HAVE TO BE PERFORMED BEFORE ANY FORMAT CONTROL OR
                                ;SKIP COUNT TESTS CAN BE DONE.
                                ;THIS TEST IS TO CHECK THE WMC SELF-DIAGNOSIS CAPABILITIES

4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WMC SELF DIAGNOSTICS
4300                            SM
 (1)                            ; ***************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
                                ;*M8959, M8957
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  CLEAR THE READ PATH MACHINE
                                ;*  SET DDR CONTROL TO 'IN'
                                ;*  SET CLOCK CONTROL TO 'NORMAL'
                                ;*  ISSUE A WMC RESTART
                                ;*  IF WMC NOT-DONE IS SET
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT ERROR-SELF-DIAGNOSTIC FAILURE
                                ;*  ENDIF
                                ;*  IF CBUSSTA HAS WMC 'LEFT' CLEAR
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC RESTART FAILED TO CLEAR 'LEFT' MODE
                                ;*  ENDIF
                                ;*  SMALL DELAY FOR WMC SELF-DIAGNOSTIC TO FINISH
                                ;*  IF WMCERR STATUS INDICATES NO ERROR BITS
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT SELF-DIAGNOSTICS DETECT ERROR
                                ;*  ENDIF
                                ;*ENDTST
4300                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*WMC1 MICRO TEST 01
                                ;*WMC1 MICRO ERROR 01
                                ;*WMC1-WMC SELF-DIAGNOSTICS
                                ;*M8959, M8957
```

```
1371                                    ;*WMC FAILED TO SET 'WMC NOT DONE' TO INDICATE SELF-DIAG. RUNNING
1372                                    ;*
1373                                    ;*WMC1 MICRO TEST 01
1374                                    ;*WMC1 MICRO ERROR 02
1375                                    ;*WMC1-WMC SELF-DIAGNOSTICS
1376                                    ;*M8959, M8957
1377                                    ;*WMC RESTART FAILED TO CLEAR THE WMC 'LEFT' BIT IN CBUSSTA WORD (241)
1378                                    ;*ACTUAL = NNNN
1379                                    ;*
1380                                    ;*WMC1 MICRO TEST 01
1381                                    ;*WMC1 MICRO ERROR 03
1382                                    ;*WMC1-WMC SELF-DIAGNOSTICS
1383                                    ;*M8959, M8957
1384                                    ;*WMC SELF-TESTS SUSPECTED OF FINDING A FAILURE! ACTUAL WORD IS THE
1385                                    ;*WMC ERROR STATUS BYTE FROM WMCERR (332)
1386                                    ;*ACTUAL = NNNN
1387  4300                              $
 (1)                                    ; ***************************************************************************
1388
1389  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01            7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1390                                    ;%WMC1-WMC SELF-DIAGNOSTICS
1391                                    ;&M8959, M8957
1392  4305  3E  10            7.0                MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1393  4307  D3  F0           10.0                OUT     CLKCTL
1394  4309  3E  10            7.0                MVI     A,W.GCR         ;SET GCR MODE
1395  430B  D3  D3           10.0                OUT     WMCCTL
1396  430D  3E  20            7.0                MVI     A,R.PLO1        ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1397  430F  D3  09           10.0                OUT     RPCTL
1398  4311  AF                4.0                XRA     A
1399  4312  D3  0A           10.0                OUT     RTIEB           ;CLEAR THE TIE BUS
1400  4314  3E  0D            7.0                MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1401  4316  D3  08           10.0                OUT     RCMD
1402  4318  00                4.0                NOP                     ;WAIT
1403  4319  00                4.0                NOP                     ;WAIT
1404  431A  00                4.0                NOP
1405  431B  00                4.0                NOP
1406  431C  00                4.0                NOP
1407  431D  3E  21            7.0                MVI     A,R.PLO1!R.STPC ;STOP THE READ PATH
1408  431F  D3  09           10.0                OUT     RPCTL
1409
1410                                    ;SET DDR CONTROL SO CORRECT ERROR INFO CAN BE FOUND
1411
1412  4321  3E  89            7.0                MVI     A,@211          ;SET DDRC TO 'IN'
1413  4323  D3  DB           10.0                OUT     DDRCTL          ;LOAD THE DDR CONTROL WORD
1414
1415                                    ;RESTART WMC
1416
1417  4325  AF                4.0  TST1L:   XRA     A                   ;SET THE SYSTEM CLOCK TO 'NORMAL'
1418  4326  D3  F0           10.0                OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
1419  4328  3E  01            7.0                MVI     A,W.RST         ;THEN RESTART THE WMC
1420  432A  D3  D3           10.0                OUT     WMCCTL
1421  432C  00                4.0                NOP                     ;SMALL DELAY
```

```
1422  432D  AF                    4.0          XRA     A                ;FINISH THE RESTART
1423  432E  D3  D3                10.0          OUT     WMCCTL           ;CLEAR THE RESTART BIT
1424  4330  DB  D0                10.0          IN      WMCSTA           ;GET WMC STATUS WORD
1425  4332  E6  40                 7.0          ANI     W.DONN           ;SAVE ONLY 'WMC NOT DONE' BIT
1426  4334  C2  3C  43            10.0          JNZ     TST1AC
1427  4337                                       ERR     TST1L,TST1AC
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4337  CD  09  28            18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0001                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  433A  01                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  433B  00                                          .BYTE
 (1)  433C  CD  15  28            18.0          TST1AC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  433F  DA  25  43            10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
1428                          ;>WMC FAILED TO SET 'WMC NOT DONE' TO INDICATE SELF-DIAG. RUNNING
1429                          ;THIS SECTION CHECKS THAT THE RESTART COMMAND RESET THE WMC 'LEFT' BIT
1430                          ;IN THE 'CBUSSTA' WORD (241(8)).
1431
1432  4342  DB  A1                 10.0  TST1B:  IN      CBUSSTA          ;GET THE STATUS DATA
1433  4344                                       ROUT    ADATA
 (1)  4344  D3  94                10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4346  7F                     4.0                  MOV     A,A              ;RETRY LINK
1434  4347  E6  04                 7.0          ANI     W.LEFT           ;SAVE ONLY THE 'LEFT' BIT
1435  4349  CA  51  43            10.0          JZ      TST1BC           ;OK IF =0
1436  434C                                       ERRA    TST1L,TST1BC
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  434C  CD  0F  28            18.0                  CALL    ERLPA            ;PROCESS ERROR - DO 2.3
 (1)        0002                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  434F  02                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4350  00                                          .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  4351  CD  15  28            18.0          TST1BC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4354  DA  25  43            10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
1437                          ;>WMC RESTART FAILED TO CLEAR THE WMC 'LEFT' BIT IN CBUSSTA WORD (241)
1438
1439  4357  3E  28                 7.0  TST1C:  MVI     A,40             ;TEST FOR ERROR DETECTED BY SELF TESTS
1440  4359  3D                     4.0  TST1W:  DCR     A                ;DOWNCOUNT THE WAIT COUNTER
1441  435A  C2  59  43            10.0          JNZ     TST1W            ;STAY HERE TILL =0
1442  435D  DB  DA                10.0          IN      WMCERR           ;GET THE WMC ERROR STATUS BYTE
1443  435F                                       ROUT    ADATA            ;SAVE FOR ERROR REPORT
 (1)  435F  D3  94                10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4361  7F                     4.0                  MOV     A,A              ;RETRY LINK
1444  4362  E6  30                 7.0          ANI     W.ERR!W.ROME     ;SAVE ONLY THE ERROR BITS
1445  4364  CA  6C  43            10.0          JZ      TST1CC           ;JUMP OVER IF NO ERROR DETECTED
```

```
1447  4367                                          ERRA    TST1L,TST1CC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4367  CD  0F  28      18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0003                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  436A  03                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  436B  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  436C  CD  15  28      18.0            TST1CC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  436F  DA  25  43      10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1448                                    ;>WMC SELF-TESTS SUSPECTED OF FINDING A FAILURE! ACTUAL WORD IS THE
1449                                    ;>WMC ERROR STATUS BYTE FROM WMCERR (332)
1450  4372                                      ENDTST  TST1L
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4372                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4372  CD  06  28      18.0                    CALL    REQST
 (2)  4375  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4376  00  00                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  4378  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  437A  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  437B  07                                      .BYTE   7               ;REQUEST CODE
 (1)  437C  3A  9A  4F      13.0                    LDA     ITERA           ;GET ITERATION COUNT
 (1)  437F  3D               4.0                    DCR     A               ;DOWNCOUNT
 (1)  4380  32  9A  4F      13.0                    STA     ITERA           ;SAVE COUNT
 (1)  4383  F2  25  43      10.0                    JP      TST1L           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - WMC BASIC SANITY CHECK
4386                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WMC BASIC SANITY CHECK
4386                            SM
                                ;*********************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959
4386                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A BASIC SANITY CHECK OF THE WRITE MICRO CONTROLLER
                                ;*BY SETTING BOTH THE 'RESTART' AND 'ENABLE WRITE MICRO CONTROLLER' BITS
                                ;*IN THE WMC CONTROL REGISTER.  THIS BIT COMBINATION SHOULD CAUSE THE
                                ;*WRITE MICRO CONTROLLER STATUS BIT 'XMC NOT DONE' TO RESET AND STAY
                                ;*RESET.
4386                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  CLEAR DATACTL BITS SO WMC WON'T HAVE INIT ERRORS
                                ;*  SET WMC RESTART + ENABLE BITS
                                ;*  CLEAR WMC RESTART BIT FINISH RESTART CYCLE
                                ;*  SMALL DELAY
                                ;*  IF WMCSTA HAS NOT-DONE SET
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC ROM PGM FAILURE
                                ;*  ENDIF
                                ;*  CLEAR WMC ENABLE BIT IN WMCCTL-CAUSE A WMC CLEAR FUNCTION
                                ;*  IF WMCERR INDICATES NO ERROR
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC RESTART FAILURE
                                ;*  ENDIF
                                ;*  IF INSTA X.ROME BIT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC RESTART FAILED TO CLEAR TRANSLATION
                                ;*  ENDIF
                                ;*  IF WMCSTA NOT-DONE BIT-1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC FAILED TO FINISH
                                ;*  ENDIF
                                ;*ENDTST
4386                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 02
                                ;*WMC1 MICRO ERROR 04
                                ;*WMC1-WMC BASIC SANITY CHECK
```

```
1491                                    ;*M8959
1492                                    ;*WMC STATUS BIT 'WMC NOT DONE'' CLEAR AFTER ENABLE/RESTART ISSUED
1493                                    ;*
1494                                    ;*WMC1 MICRO TEST 02
1495                                    ;*WMC1 MICRO ERROR 05
1496                                    ;*WMC1-WMC BASIC SANITY CHECK
1497                                    ;*M8959
1498                                    ;*WMC RESTART DID NOT CLEAR THE WMC ERRORS BYTE
1499                                    ;*ACTUAL = NNNN
1500                                    ;*
1501                                    ;*WMC1 MICRO TEST 02
1502                                    ;*WMC1 MICRO ERROR 06
1503                                    ;*WMC1-WMC BASIC SANITY CHECK
1504                                    ;*M8959
1505                                    ;*WMC RESTART DID NOT CLEAR THE XMC ROM PARITY ERRORS BIT
1506                                    ;*ACTUAL = NNNN
1507                                    ;*
1508                                    ;*WMC1 MICRO TEST 02
1509                                    ;*WMC1 MICRO ERROR 07
1510                                    ;*WMC1-WMC BASIC SANITY CHECK
1511                                    ;*M8959
1512                                    ;*WMC STATUS BIT 'WMC NOT DONE' NOT SET AFTER 'ENABLE'+'RESTART' DROPPED
1513  4386                              S
 (1)                                    ; *****************************************************************************
1514  4386                              TEST2:  TESTX   @2
 (1)  4386  3E  02            7.0                MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4388  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1515                                    ;%WMC1-WMC BASIC SANITY CHECK
1516                                    ;&M8959
1517  438B  AF                  4.0   TST2L:  XRA     A               ;CLEAR THE FORMAT BITS
1518  438C  D3  D0              10.0          OUT     DATACTL         ;SO WMC WON'T HAVE ERROR WHEN STARTED
1519  438E  3E  8*               7.0          MVI     A,W.RST.W.ENAB  ;SET 'EN WMC' AND 'RESTART'
1520  4390  D3  D3              10.0          OUT     WMCCTL          ;
1521  4392  3E  80               7.0          MVI     A,W.ENAB        ;CLEAR THE RESTART BIT
1522  4394  D3  D3              10.0          OUT     WMCCTL
1523  4396  00                  4.0          NOP                     ;DELAY
1524  4397  00                  4.0          NOP                     ;DELAY
1525  4398  DB  D0              10.0          IN      WMCSTA          ;GET WMC STATUS BYTE
1526  439A  E6  40               7.0          ANI     W.DONN          ;CHECK FOR 'WMC NOT DONE'
1527  439C  C2  A4  43          10.0          JNZ     TST2AC          ;CONTINUE IF SET
1528  439F                              ERR     TST2L,TST2AC
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  439F  CD  09  28       18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43A2  04                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43A3  00                               .BYTE
 (1)  43A4  CD  15  28       18.0    TST2AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43A7  DA  88  43       10.0                JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1529                                ;>WMC STATUS BIT 'WMC NOT DONE'' CLEAR AFTER ENABLE/RESTART ISSUED
1530  43AA  AF                  4.0          XRA     A               ;NOW DROP ENABLE TO CAUSE A CLEAR FUNCTION
1531  43AB  D3  D3              10.0          OUT     WMCCTL          ;IN THE WMC ROM PROGRAM
1532  43AD  DB  DA              10.0          IN      WMCERR          ;GET THE WMC ERRORS BYTE
1533  43AF  E6  78               7.0          ANI     M.PE.M.RDPE.W.ROME.W.ERR
1534  43B1  CA  BC  43          10.0          JZ      TST2BC          ;JUMP IF NO ERROR BIT UP
```

```
1535  4384                                          ROUT    ADATA
 (1)  4384  D3  94          10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4386  7F               4.0                   MOV     A,A             ;RETRY LINK
1536  4387                                          ERRA    TST2L,TST2BC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA ' VALID
 (1)  4387  CD  0F  28      18.0                           CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0005                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  438A  05                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  438B  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  438C  CD  15  28      18.0                   TST2BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  438F  DA  8B  43      10.0                           JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1537                                            ;>WMC RESTART DID NOT CLEAR THE WMC ERRORS BYTE
1538  43C2  DB  E0            10.0                   IN      INTSTA          ;GET THE INTERRUPT STATUS WORD
1539  43C4  E6  01             7.0                   ANI     X.ROME
1540  43C6                                          ROUT    ADATA
 (1)  43C6  D3  94            10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43C8  7F                 4.0                   MOV     A,A             ;RETRY LINK
1541  43C9  CA  D1  43         10.0                   JZ      TST2CC
1542  43CC                                          ERRA    TST2L,TST2CC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  43CC  CD  0F  28      18.0                           CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0006                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43CF  06                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43D0  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43D1  CD  15  28      18.0                   TST2CC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43D4  DA  8B  43      10.0                           JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1543                                            ;>WMC RESTART DID NOT CLEAR THE XMC ROM PARITY ERRORS BIT
1544
1545  43D7  00                  4.0             TST2D:  NOP
1546  43D8  DB  D0            10.0                   IN      WMCSTA          ;GET WMC STATUS WORD
1547  43DA  E6  40             7.0                   ANI     W.DONN          ;SAVE ONLY 'WMC NOT DONE' BIT
1548  43DC  C2  E4  43         10.0                   JNZ     TST2DC
1549  43DF                                          ERR     TST2L,TST2DC
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43DF  CD  09  28      18.0                           CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                         MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E2  07                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E3  00                                           .BYTE
 (1)  43E4  CD  15  28      18.0                   TST2DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E7  DA  8B  43      10.0                           JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1550                                            ;>WMC STATUS BIT 'WMC NOT DONE' NOT SET AFTER 'ENABLE' + 'RESTART' DROPPED
1551  43EA                                          ENDTST  TST2L
 (1)                                              ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43EA                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43EA  CD  06  28      18.0                           CALL    REQST
 (2)  43ED  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  43EE  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  43F0  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43F2  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43F3  07                                           .BYTE   7               ;REQUEST CODE
 (1)  43F4  3A  9A  4F      13.0                   LDA     ITERA           ;GET ITERATION COUNT
 (1)  43F7  3D               4.0                   DCR     A               ;DOWNCOUNT
 (1)  43F8  32  9A  4F      13.0                   STA     ITERA           ;SAVE COUNT
 (1)  43FB  F2  8B  43      10.0                   JP      TST2L           ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 3 - WMC RE-INITIALIZE TEST
1554  43FE                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1555                            ;*WMC RE-INITIALIZE TEST
1556  43FE                      SM
 (1)                            ;*********************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
1557                            ;*M8959
1558  43FE                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1559                            ;*THIS TEST IS CHECKING THAT THE WMC STATUS WORD (100320)
1560                            ;*DOES NOT INDICATE 'WMC NOT READY' STATUS AFTER THE WMC CONTROL
1561                            ;*WORD (100323) IS LOADED WITH A RESTART COMMAND - A RESTART WILL
1562                            ;*CAUSE THE WMC ROM PROGRAM TO RE-INITIALIZE TO A KNOWN STATE.
1563  43FE                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1564                            ;*BGNTST
1565                            ;*  SET WMCCTL RESTART BIT
1566                            ;*  IF WMCERR INDICATES ERROR
1567                            ;*  :  THEN-REPORT ERROR AFTER WMC RESTART SET
1568                            ;*  :  ELSE-CONTINUE
1569                            ;*  ENDIF
1570                            ;*  FINISH WMC RESTART CYCLE
1571                            ;*  IF WMCERR INDICATES ERROR
1572                            ;*  :  THEN-REPORT ERROR AFTER WMC RESTART CLEARED
1573                            ;*  :  ELSE-CONTINUE
1574                            ;*  ENDIF
1575                            ;*ENDTST
1576  43FE                      SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1577                            ;*WMC1 MICRO TEST 03
1578                            ;*WMC1 MICRO ERROR 10
1579                            ;*WMC1-WMC RE-INITIALIZE TEST
1580                            ;*M8959
1581                            ;*WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' SET
1582                            ;*
1583                            ;*WMC1 MICRO TEST 03
1584                            ;*WMC1 MICRO ERROR 11
1585                            ;*WMC1-WMC RE-INITIALIZE TEST
1586                            ;*M8959
1587                            ;*WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' CLEARED
1588  43FE                      S
 (1)                            ;*********************************************************************
```

```
1590  43FE                                TEST3:  TESTX   @3
 (1)  43FE  3E  03              7.0                      MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  4400  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
1591                                    ;%WMC  WMC RE-INITIALIZE TEST
1592                                    ;&M8, ,
1593
1594                                    ;ISSUE A 'RESTART' COMMAND TO WMC CONTROL WORD
1595
1596  4403  3E  81              7.0  TST3L:  MVI     A,W.ENAB.W.RST          ;GET THE RESTART BIT
1597  4405  D3  D3             10.0           OUT     WMCCTL          ;ISSUE THE COMMAND
1598
1599                                    ;WMC ROM ERROR TEST
1600
1601  4407  DB  DA             10.0  TST3A:  IN      WMCERR          ;GET THE WMC ERROR WORD
1602  4409  E6  30              7.0           ANI     W.ROME.W.ERR    ;SAVE 'WMC ROM ERR' & 'ERROR'
1603  440B  CA  13  44         10.0           JZ      TST3AC          ;JUMP OVER IF =0 - NO ERROR SET
1604  440E                                   ERR     TST3L,TST3AC
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  440E  CD  09  28         18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4411  08                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4412  00                                           .BYTE
 (1)  4413  CD  15  28         18.0                TST3AC::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4416  DA  03  44         10.0                      JC      TST3L           ;LOOP ADDRESS IF LOOP SPECIFIED
1605                                    ;>WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' SET
1606  4419  3E  80              7.0           MVI     A,W.ENAB        ;FINISH THE RESTART CYCLE
1607  441B  D3  D3             10.0           OUT     WMCCTL          ;CLEAR RESTART BIT TO FINISH THE RESTART
1608
1609                                    ;WMC ROM ERROR TEST
1610  441D  DB  DA             10.0  TST3B:  IN      WMCERR          ;GET THE WMC ERROR WORD
1611  441F  E6  30              7.0           ANI     W.ROME!W.ERR    ;SAVE 'WMC ROM ERR' & 'ERROR'
1612  4421  CA  29  44         10.0           JZ      TST3BC          ;JUMP OVER IF =0 - NO ERROR SET
1613  4424                                   ERR     TST3L,TST3BC
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4424  CD  09  28         18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0009                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4427  09                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4428  00                                           .BYTE
 (1)  4429  CD  15  28         18.0                TST3BC::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  442C  DA  03  44         10.0                      JC      TST3L           ;LOOP ADDRESS IF LOOP SPECIFIED
1614                                    ;>WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' CLEARED
1615  442F                                     ENDTST  TST3L
 (1)                                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  442F                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  442F  CD  06  28         18.0                      CALL    REQST
 (2)  4432  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4433  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  4435  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4437  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4438  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4439  3A  9A  4F         13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  443C  3D                  4.0           DCR     A               ;DOWNCOUNT
 (1)  443D  32  9A  4F         13.0           STA     ITERA           ;SAVE COUNT
 (1)  4440  F2  03  44         10.0           JP      TST3L           ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 4 - BYTE COUNTER TEST
1618  4443                      ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*BYTE COUNTER TEST
1620  4443                      SM
                                ;*****************************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959
1622  4443                      SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE BYTE COUNTERS ABILITY TO BE LOADED WITH ALL 0'S
                                ;*AND ALL 1'S
1625  4443                      SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   ISSUE A WMC RESTART
                                ;*   SET WMCCTL 'DIAG' BIT
                                ;*   SET WMCCTL 'DIAG' + 'ENABLE'-ENTER SPECIAL WMC ROM CODE
                                ;*   SMALL DELAY FOR WMC TO CLOCK THE 8253 COUNTER CHIP
                                ;*   LOAD WMC BYTE COUNTER WITH 0'S
                                ;*   IF BYTECT=0
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-REPORT 0'S LOAD FAILURE
                                ;*   ENDIF
                                ;*   LOAD BYTE COUNTER WITH 177777
                                ;*   IF BYTECT=ALL 1'S
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-REPORT 1'S LOAD FAILURE
                                ;*   ENDIF
                                ;*ENDTST
1642  4443                      SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 04
                                ;*WMC1 MICRO ERROR 12
                                ;*WMC1-WMC BYTE COUNTER TESTS
                                ;*M8959
                                ;*WMC BYTE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
                                ;*
                                ;*WMC1 MICRO TEST 04
                                ;*WMC1 MICRO ERROR 13
                                ;*WMC1-WMC BYTE COUNTER TESTS
                                ;*M8959
                                ;*WMC BYTE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
```

```
                                        ;*
                                        ;*WMC1 MICRO TEST 04
                                        ;*WMC1 MICRO ERROR 14
                                        ;*WMC1-WMC BYTE COUNTER TESTS
                                        ;*M8959
                                        ;*WMC BYTE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 04
                                        ;*WMC1 MICRO ERROR 15
                                        ;*WMC1-WMC BYTE COUNTER TESTS
                                        ;*M8959
                                        ;*WMC BYTE COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL - NNNN
4443                                    S
 (1)                                    ; ************************************************************************************
                                        TEST4:  TESTX   @4
 (1) 4443  3E  04            7.0                    MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1) 4445  CD  03  28       18.0                    CALL    TSET            ;SETUP THE TEST
                                        ;%WMC1-WMC BYTE COUNTER TESTS
                                        ;&M8959
                                        ;INIT THE WRITE MICRO TO ENTER A SPECIAL LOOP TO JUST 'FEED' THE COUNTER
                                        ;CHIP (INTEL 8253) FOR BYTE COUNTER, PAD COUNTER, AND ECODE COUNTER.

4448  3E  01            7.0   TST4ST: MVI     A,W.RST         ;RESTART THE WRITE MICRO
444A  D3  D3           10.0           OUT     WMCCTL
444C  3E  02            7.0           MVI     A,W.DIAG
444E  D3  D3           10.0           OUT     WMCCTL
4450  3E  82            7.0           MVI     A,W.ENAB!W.DIAG ;FEED CLOCKS TO THE COUNTER CHIP
4452  D3  D3           10.0           OUT     WMCCTL
4454  3E  0A            7.0           MVI     A,10            ;GET A 'STALL' TIME
4456  3D                4.0   T4W:    DCR     A
4457  C2  56  44       10.0           JNZ     T4W             ;WAIT TILL COUNT TO 0

                                        ;LOAD WMC BYTE COUNTER WITH ALL 0'S

445A  3E  30            7.0   TST4L:  MVI     A,@60           ;SELECT THE BYTE COUNTER IN COUNTER
445C  D3  D7           10.0           OUT     CNTCTL          ;CONTROL WORD
445E  3E  00            7.0           MVI     A,0             ;THEN LOAD THE COUNTER WITH 0'S
4460  D3  D4           10.0           OUT     BYTCNT          ;LOAD BITS 7-0
4462  D3  D4           10.0           OUT     BYTCNT          ;LOAD BITS 15-8
4464  00                4.0           NOP
4465  DB  D4           10.0           IN      BYTCNT          ;GET BITS 7-0 FROM COUNTER
4467  47                4.0           MOV     B,A             ;TEMP SAVE THE DATA
4468                                    ROUT    ADATA           ;SAVE AS 'ACTUAL'
 (1) 4468  D3  94       10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 446A  7F            4.0                   MOV     A,A             ;RETRY LINK
446B  DB  D4           10.0           IN      BYTCNT          ;GET BITS 15-8
446D  4F                4.0           MOV     C,A             ;TEMP SAVE THEM
446E  78                4.0           MOV     A,B             ;RESET REG A TO BITS 7-0
446F  FE  00            7.0           CPI     0               ;TEST FOR 0'S DATA BYTE
4471  CA  79  44       10.0           JZ      TST4AC          ;JUMP IF =0 - ALL OK
```

```
1705  4474                                          ERRA    TST4L,TST4AC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4474  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000A                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4477  0A                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4478  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4479  CD  15  28        18.0                  TST4AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  447C  DA  5A  44        10.0                          JC      TST4L           ;LOOP ADDRESS IF LOOP SPECIFIED
1706                                          ;>WMC BYTE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
1707
1708  447F  79                   4.0  TST4B:  MOV     A,C             ;SET REG A = BITS 15-8
1709  4480                              ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 (1)  4480  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4482  7F                  4.0                  MOV     A,A             ;RETRY LINK
1710  4483  FE  00              7.0          CPI     0               ;TEST FOR 0'S
1711  4485  CA  8D  44         10.0          JZ      TST4BC          ;JUMP IF =0 - ALL OK
1712  4488                              ERRA    TST4L,TST4BC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4488  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000B                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448B  0B                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  448C  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  448D  CD  15  28        18.0                  TST4BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4490  DA  5A  44        10.0                          JC      TST4L           ;LOOP ADDRESS IF LOOP SPECIFIED
1713                                          ;>WMC BYTE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1714  4493  3E  30              7.0  TST4C:  MVI     A,@60           ;SELECT BYTE COUNTER IN COUNTER CONTROL
1715  4495  D3  D7             10.0          OUT     CNTCTL
1716  4497  3E  FF              7.0          MVI     A,@377          ;AND WRITE THE COUNTER TO ALL 1'S
1717  4499  D3  D4             10.0          OUT     BYTCNT          ;LOAD BITS 7-0
1718  449B  D3  D4             10.0          OUT     BYTCNT          ;LOAD BITS 15-8
1719  449D  00                  4.0          NOP
1720  449E  DB  D4             10.0          IN      BYTCNT          ;GET BITS 7-0 FROM COUNTER
1721  44A0  47                  4.0          MOV     B,A             ;TEMP SAVE THEM
1722  44A1                              ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  44A1  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44A3  7F                  4.0                  MOV     A,A             ;RETRY LINK
1723  44A4  DB  D4             10.0          IN      BYTCNT          ;GET BITS 15-8 FROM COUNTER
1724  44A6  4F                  4.0          MOV     C,A             ;TEMP SAVE THEM
1725
1726                                          ;TEST BYTE COUNTER FOR ALL 1'S NOW
1727  44A7  78                  4.0          MOV     A,B             ;SET REG A = BITS 7-0
1728  44A8  FE  FF              7.0          CPI     @377            ;ALL 1'S?
1729  44AA  CA  B2  44         10.0          JZ      TST4CA          ;JUMP IF OK
1730  44AD                              ERRA    TST4C,TST4CA
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  44AD  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000C                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B0  0C                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B1  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44B2  CD  15  28        18.0                  TST4CA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44B5  DA  93  44        10.0                          JC      TST4C           ;LOOP ADDRESS IF LOOP SPECIFIED
1731                                          ;>WMC BYTE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1732
1733  44B8  79                   4.0  TST4D:  MOV     A,C             ;GET BITS 15-8
```

```
1734  44B9                                        ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  44B9  D3  94            10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44BB  7F                 4.0                       MOV     A,A             ;RETRY LINK
1735  44BC  FE  FF               7.0               CPI     @377            ;ALL 1'S?
1736  44BE  CA  C6  44          10.0               JZ      TST4DC          ;JUMP IF ALL 1'S
1737  44C1                                        ERRA    TST4C,TST4DC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  44C1  CD  0F  28          18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000D                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C4  0D                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44C5  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44C6  CD  15  28          18.0               TST4DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44C9  DA  93  44          10.0                       JC      TST4C           ;LOOP ADDRESS IF LOOP SPECIFIED
1738                                        ;>WMC BYTE COUNTER BITS 15-8 LOADED WITH ALL 1'S BU" JOT 1'S WHEN READ
1739  44CC                                        ENDTST  TST4ST
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44CC                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44CC  CD  06  28          18.0                       CALL    REQST
 (2)  44CF  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  44D0  00  00                                       .WORD                   ;SYSTEM "" COUNT
 (2)  44D2  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44D4  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44D5  07                                           .BYTE   7               ;REQUEST CODE
 (1)  44D6  3A  9A  4F          13.0               LDA     ITERA           ;GET ITERATION COUNT
 (1)  44D9  3D                  4.0               DCR     A               ;DOWNCOUNT
 (1)  44DA  32  9A  4F          13.0               STA     ITERA           ;SAVE COUNT
 (1)  44DD  F2  48  44          10.0               JP      TST4ST          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 5 - WMC PAD COUNTER TEST
                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PAD COUNTER TEST
                                SM
                                ;*****************************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE PAD COUNTERS ABILITY TO BE LOADED WITH ALL 0'S AND
                                ;*ALL 1'S
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  SET WMCCTL RESTART BIT
                                ;*  IF WMCERR INDICATES ERROR
                                ;*  :  THEN-REPORT ERROR AFTER WMC RESTART SET
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  FINISH WMC RESTART CYCLE
                                ;*  IF WMCERR INDICATES ERROR
                                ;*  :  THEN-REPORT ERROR AFTER WMC RESTART CLEARED
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*ENDTST
                                SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 05
                                ;*WMC1 MICRO ERROR 16
                                ;*WMC1-WMC PAD COUNTER TESTS
                                ;*M8959
                                ;*WMC PAD COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
                                ;*
                                ;*WMC1 MICRO TEST 05
                                ;*WMC1 MICRO ERROR 17
                                ;*WMC1-WMC PAD COUNTER TESTS
                                ;*M8959
                                ;*WMC PAD COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
                                ;*
                                ;*WMC1 MICRO TEST 05
                                ;*WMC1 MICRO ERROR 20
                                ;*WMC1-WMC PAD COUNTER TESTS
```

```
                                        ;*M8959
                                        ;*WMC PAD COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 05
                                        ;*WMC1 MICRO ERROR 21
                                        ;*WMC1-WMC PAD COUNTER TESTS
                                        ;*M8959
                                        ;*WMC PAD COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
                                        S
 44E0                                   ; ***************************************************************************
  (1)
                                        TEST5:  TESTX   @5
 44E0                             7.0                   MVI     A,@5            ;DEFINE THE TEST NUMBER
 44E0  3E  05                    18.0                   CALL    TSET            ;SETUP THE TEST
 44E2  CD  03  28
                                        ;%WMC1-WMC PAD COUNTER TESTS
                                        ;&M8959
                                        ;LOAD WMC PAD COUNTER WITH ALL 0'S

 44E5  3E  70                     7.0   TST5L:  MVI     A,@160          ;SELECT THE PAD COUNTER IN COUNTER
 44E7  D3  D7                    10.0           OUT     CNTCTL          ;CONTROL WORD
 44E9  3E  00                     7.0           MVI     A,0             ;THEN LOAD THE COUNTER WITH 0'S
 44EB  D3  D5                    10.0           OUT     PADCNT          ;LOAD BITS 7-0
 44ED  D3  D5                    10.0           OUT     PADCNT          ;LOAD BITS 15-8
 44EF  00                         4.0           NOP
 44F0  DB  D5                    10.0           IN      PADCNT          ;GET BITS 7-0 FROM COUNTER
 44F2  47                         4.0           MOV     B,A             ;TEMP SAVE THE DATA
 44F3                                           ROUT    ADATA           ;SAVE AS 'ACTUAL'
 44F3  D3  94                    10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 44F5  7F                         4.0                   MOV     A,A             ;RETRY LINK
 44F6  DB  D5                    10.0           IN      PADCNT          ;GET BITS 15-8
 44F8  4F                         4.0           MOV     C,A             ;TEMP SAVE THEM
 44F9  78                         4.0           MOV     A,B             ;RESET REG A TO BITS 7-0
 44FA  FE  00                     7.0           CPI     0               ;TEST FOR 0'S DATA BYTE
 44FC  CA  04  45                10.0           JZ      TST5AC          ;JUMP IF =0 - ALL OK
 44FF                                           ERRA    TST5L,TST5AC
  (1)                                           ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 44FF  CD  0F  28                18.0                   CALL    ERLPA           ;PROCESS ERROR - DO 2.3
       000E                                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 4502  0E                                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 4503  00                                               .BYTE                   ;PRINT ROUTINE NUMBER
 4504  CD  15  28                18.0           TST5AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 4507  DA  E5  44                10.0                   JC      TST5L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>WMC PAD COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                        ;CHECK BITS 15-0 FOR 0'S

 450A  79                         4.0   TST5B:  MOV     A,C             ;SET REG A = BITS 15-8
 450B                                           ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 450B  D3  94                    10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 450D  7F                         4.0                   MOV     A,A             ;RETRY LINK
 450E  FE  00                     7.0           CPI     0               ;TEST FOR 0'S
 4510  CA  18  45                10.0           JZ      TST5BC          ;JUMP IF =0 - ALL OK
```

```
1820  4513                                          ERRA    TST5L,TST5BC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4513  CD  0F  28      18.0                            CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000F                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4516  0F                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4517  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4518  CD  15  28      18.0                TST5BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  451B  DA  E5  44      10.0                            JC      TST5L           ;LOOP ADDRESS IF LOOP SPECIFIED
1821                                        ;>WMC PAD COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1822  451E  3E  70               7.0       TST5C:  MVI     A,@160          ;SELECT PAD COUNTER IN COUNTER CONTROL
1823  4520  D3  D7              10.0               OUT     CNTCTL
1824  4522  3E  FF               7.0               MVI     A,@377          ;AND WRITE THE COUNTER TO ALL 1'S
1825  4524  D3  D5              10.0               OUT     PADCNT          ;LOAD BITS 7-0
1826  4526  D3  D5              10.0               OUT     PADCNT          ;LOAD BITS 15-8
1827  4528  00                   4.0               NOP
1828  4529  DB  D5              10.0               IN      PADCNT          ;GET BITS 7-0 FROM COUNTER
1829  452B  47                   4.0               MOV     B,A             ;TEMP SAVE THEM
1830  452C                                         ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  452C  D3  94              10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  452E  7F                   4.0                       MOV     A,A             ;RETRY LINK
1831  452F  DB  D5              10.0               IN      PADCNT          ;GET BITS 15-8 FROM COUNTER
1832  4531  4F                   4.0               MOV     C,A             ;TEMP SAVE THEM
1833
1834                                        ;TEST PAD COUNTER FOR ALL 1'S NOW
1835
1836  4532  78                   4.0               MOV     A,B             ;SET REG A = BITS 7-0
1837  4533  FE  FF               7.0               CPI     @377            ;ALL 1'S?
1838  4535  CA  3D  45          10.0               JZ      TST5CA          ;JUMP IF OK
1839  4538                                         ERRA    TST5C,TST5CA
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4538  CD  0F  28      18.0                            CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0010                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453B  10                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453C  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  453D  CD  15  28      18.0                TST5CA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4540  DA  1E  45      10.0                            JC      TST5C           ;LOOP ADDRESS IF LOOP SPECIFIED
1840                                        ;>WMC PAD COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1841  4543  79                   4.0       TST5D:  MOV     A,C             ;GET BITS 15-8
1842  4544                                         ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  4544  D3  94              10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4546  7F                   4.0                       MOV     A,A             ;RETRY LINK
1843  4547  FE  FF               7.0               CPI     @377            ;ALL 1'S?
1844  4549  CA  51  45          10.0               JZ      TST5DC          ;JUMP IF ALL 1'S
1845  454C                                         ERRA    TST5C,TST5DC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  454C  CD  0F  28      18.0                            CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0011                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454F  11                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4550  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4551  CD  15  28      18.0                TST5DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4554  DA  1E  45      10.0                            JC      TST5C           ;LOOP ADDRESS IF LOOP SPECIFIED
1846                                        ;>WMC PAD COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
```

```
1848  4557                                          ENDTST  TST5L
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4557                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4557  CD  06  28        18.0                                  CALL    REQST
 (2)  455A  00                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  455B  00  00                                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  455D  00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455F  00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4560  07                                                      .BYTE   7               ;REQUEST CODE
 (1)  4561  3A  9A  4F        13.0                  LDA     ITERA           ;GET ITERATION COUNT
 (1)  4564  3D                 4.0                  DCR     A               ;DOWNCOUNT
 (1)  4565  32  9A  4F        13.0                  STA     ITERA           ;SAVE COUNT
 (1)  4568  F2  E5  44        10.0                  JP      TST5L           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 6 - WMC ECODE COUNTER TEST
                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ECODE COUNTER TEST
                                SM
                                ;*********************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ECODE COUNTERS ABILITY TO BE LOADED WITH ALL 0'S AND
                                ;*ALL 1'S
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  SET WMCCTL RESTART BIT
                                ;*  IF WMCERR INDICATES ERROR
                                ;*  :  THEN-REPORT ERROR AFTER WMC RESTART SET
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  FINISH WMC RESTART CYCLE
                                ;*  IF WMCERR INDICATES ERROR
                                ;*  :  THEN-REPORT ERROR AFTER WMC RESTART CLEARED
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*ENDTST
                                SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 06
                                ;*WMC1 MICRO ERROR 22
                                ;*WMC1-WMC ECODE COUNTER TESTS
                                ;*M8959
                                ;*WMC ECODE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
                                ;*
                                ;*WMC1 MICRO TEST 06
                                ;*WMC1 MICRO ERROR 23
                                ;*WMC1-WMC ECODE COUNTER TESTS
                                ;*M8959
                                ;*WMC ECODE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
                                ;*
                                ;*WMC1 MICRO TEST 06
                                ;*WMC1 MICRO ERROR 24
                                ;*WMC1-WMC ECODE COUNTER TESTS
```

```
                                        ;*M8959
                                        ;*WMC ECODE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 06
                                        ;*WMC1 MICRO ERROR 25
                                        ;*WMC1-WMC ECODE COUNTER TESTS
                                        ;*M8959
                                        ;*WMC ECODE COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
456B                                    S
 (1)                                    ; ***************************************************************************
                                        TEST6:  TESTX   @6
456B  3E  06            7.0                             MVI     A,@6            ;DEFINE THE TEST NUMBER
456D  CD  03  28       18.0                             CALL    TSET            ;SETUP THE TEST
                                        ;%WMC1-WMC ECODE COUNTER TESTS
                                        ;&M8959
                                        ;LOAD WMC ECODE COUNTER WITH ALL 0'S

4570  3E  B0            7.0  TST6L:  MVI     A,@260          ;SELECT THE ECODE COUNTER IN COUNTER
4572  D3  D7           10.0          OUT     CNTCTL          ;CONTROL WORD
4574  3E  00            7.0          MVI     A,0             ;THEN LOAD THE COUNTER WITH 0'S
4576  D3  D6           10.0          OUT     ERRCNT          ;LOAD BITS 7-0
4578  D3  D6           10.0          OUT     ERRCNT          ;LOAD BITS 15-8
457A  00               4.0          NOP
457B  DB  D6           10.0          IN      ERRCNT          ;GET BITS 7-0 FROM COUNTER
457D  47               4.0          MOV     B,A             ;TEMP SAVE THE DATA
457E                                ROUT    ADATA           ;SAVE AS 'ACTUAL'
 (1)  457E  D3  94      10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4580  7F          4.0                   MOV     A,A             ;RETRY LINK
4581  DB  D6           10.0          IN      ERRCNT          ;GET BITS 15-8
4583  4F               4.0          MOV     C,A             ;TEMP SAVE THEM
4584  78               4.0          MOV     A,B             ;RESET REG A TO BITS 7-0
4585  FE  00           7.0          CPI     0               ;TEST FOR 0'S DATA BYTE
4587  CA  8F  45      10.0          JZ      TST6AC          ;JUMP IF =0 - ALL OK
458A                                ERRA    TST6L,TST6AC
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  458A  CD  0F  28  18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0012                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  458D  12                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  458E  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  458F  CD  15  28  18.0            TST6AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4592  DA  70  45  10.0                    JC      TST6L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>WMC ECODE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                ;CHECK BITS 15-0 FOR 0'S

4595  79               4.0  TST6B:  MOV     A,C             ;SET REG A = BITS 15-8
4596                                ROUT    ADATA           ;SAVE 'ACTUAL' DATA
4596  D3  94           10.0          OUT     ADATA           ;WRITE AC INTO ADATA
4598  7F               4.0          MOV     A,A             ;RETRY LINK
4599  FE  00           7.0          CPI     0               ;TEST FOR 0'S
459B  CA  A3  45      10.0          JZ      TST6BC          ;JUMP IF -0 - ALL OK
```

```
1929  459E                                          ERRA    TST6L,TST6BC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  459E  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0013                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45A1  13                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45A2  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45A3  CD  15  28        18.0                  TST6BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45A6  DA  70  45        10.0                          JC      TST6L           ;LOOP ADDRESS IF LOOP SPECIFIED
1930                                        ;>WMC ECODE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1931  45A9  3E  B0                7.0     TST6C:  MVI     A,@260          ;SELECT ECODE COUNTER IN COUNTER CONTROL
1932  45AB  D3  D7               10.0             OUT     CNTCTL
1933  45AD  3E  FF                7.0             MVI     A,@377          ;AND WRITE THE COUNTER TO ALL 1'S
1934  45AF  D3  D6               10.0             OUT     ERRCNT          ;LOAD BITS 7-0
1935  45B1  D3  D6               10.0             OUT     ERRCNT           LOAD BITS 15-8
1936  45B3  00                    4.0             NOP
1937  45B4  DB  D6               10.0             IN      ERRCNT          ;GET BITS 7-0 FROM COUNTER
1938  45B6  47                    4.0             MOV     B,A             ;TEMP SAVE THEM
1939  45B7                                        ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  45B7  D3  94               10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45B9  7F                    4.0                     MOV     A,A             ;RETRY LINK
1940  45BA  DB  D6               10.0             IN      ERRCNT          ;GET BITS 15-8 FROM COUNTER
1941  45BC  4F                    4.0             MOV     C,A             ;TEMP SAVE THEM
1942
1943                                        ;TEST ECODE COUNTER FOR ALL 1'S NOW
1944  45BD  78                    4.0             MOV     A,B             ;SET REG A = BITS 7-0
1945  45BE  FF  FF                7.0             CPI     @377            ;ALL 1'S?
1946  45C0  CA  C8  45            10.0            JZ      TST6CA          ;JUMP IF OK
1947  45C3                                        ERRA    TST6C,TST6CA
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  45C3  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0014                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45C6  14                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45C7  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45C8  CD  15  28        18.0                  TST6CA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45CB  DA  A9  45        10.0                          JC      TST6C           ;LOOP ADDRESS IF LOOP SPECIFIED
1948                                        ;>WMC ECODE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1949
1950  45CE  79                    4.0     TST6D:  MOV     A,C             ;GET BITS 15-8
1951  45CF  D3  94               10.0             OUT     @224            ;SAVE 'ACTUAL' DATA BYTE
1952  45D1  FE  FF                7.0             CPI     @377            ;ALL 1'S?
1953  45D3  CA  DB  45            10.0            JZ      TST6DC          ;JUMP IF ALL 1'S
1954  45D6                                        ERRA    TST6C,TST6DC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  45D6  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0015                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D9  15                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45DA  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45DB  CD  15  28        18.0                  TST6DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45DE  DA  A9  45        10.0                          JC      TST6C           ;LOOP ADDRESS IF LOOP SPECIFIED
1955                                        ;>WMC ECODE COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
```

```
1957  45E1                                          ENDTST  TST6L
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45E1                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45E1  CD  06  28        18.0                                  CALL    REQST
 (2)  45E4  00                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  45E5  00  00                                                  .WORD                   ;SYSTEM '**' COUNT
 (2)  45E7  00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45E9  00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  45EA  07                                                      .BYTE   7               ;REQUEST CODE
 (1)  45EB  3A  9A  4F        13.0                  LDA     ITERA           ;GET ITERATION COUNT
 (1)  45EE  3D                 4.0                  DCR     A               ;DOWNCOUNT
 (1)  45EF  32  9A  4F        13.0                  STA     ITERA           ;SAVE COUNT
 (1)  45F2  F2  70  45        10.0                  JP      TST6L           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 7 - WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
45F5                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ILLEGAL PACKING MODE TEST
45F5                            SM
                                ;*****************************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959
45F5                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST IS ATTEMPTING TO SEE IF THE WMC ROM PROGRAM CAN DETECT
                                ;*WRONG INFORMATION IN THE WMC DATA CONTROL WORD FOR THE 'FORMAT'
                                ;*DATA...AN ILLEGAL FORMAT CODE IS WRITTEN IN THE DATA CONTROL WORD
                                ;*AND THEN SET 'WMC ENABLE' IN THE WMC CONTROL WORD TO ALLOW THE
                                ;*WMC ROM PROGRAM DETECT THE ERROR CONDITION.
                                ;*
                                ;*FORMAT CODE 7 SHOULD CAUSE THE 'ECODE' COUNTER TO CONTAIN A
                                ;*-3 COUNT WHEN THE ERROR IS DETECTED. WMC ERROR STATUS WORD SHOULD
                                ;*HAVE 'ERROR' SET ON THESE CODES.
                                ;*
                                ;*FORMAT CODES 0-6 SHOULD CAUSE NO ERROR.
45F5                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  ISSUE A WMC RESTART
                                ;*  CLEAR ERROR CODE COUNTER
                                ;*  LOAD DATACTL WITH INVALID PACKING MODE #7
                                ;*  SET WMCCTL 'ENABLE' BIT OT START WMC ROM PROGRAM PROCESS
                                ;*  SMALL DELAY SO WMC CAN DETECT ILLEGAL FORMAT NUMBER
                                ;*  IF WMCSTA NOT-DONE=0 (WMC FINISHED)
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT POSSIBLE WMC SELF-DIAG. FAILURE
                                ;*  ENDIF
                                ;*  IF ERROR COUNTER DECREMENTED TO -3 (ILLEG. FMT CODE ISSUED)
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT 'ECODE' COUNT NOT--3
                                ;*  ENDIF
                                ;*  IF WMCERR WMC ROM ERROR BIT-1 AND 'ERROR'=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT FMT CODE 7 FAILED TO CAUSE PROPER WMC ERROR STATUS
                                ;*  ENDIF
                                ;*ENDTST
45F5                            SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 07
```

```
                                        ;*WMC1 MICRO ERROR 26
                                        ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
                                        ;*M8959
                                        ;*WMC NOT READY - WMC GROSS ERROR! POSSIBLE WMC SELF-DIAGNOSIS FAILURE.
                                        ;*
                                        ;*WMC1 MICRO TEST 07
                                        ;*WMC1 MICRO ERROR 27
                                        ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
                                        ;*M8959
                                        ;*WMC 'ECODE' COUNT NOT -3 (375) AFTER ILLEGAL FORMAT CODE #7 USED
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 07
                                        ;*WMC1 MICRO ERROR 30
                                        ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
                                        ;*M8959
                                        ;*WMC 'ECODE' COUNTER BITS 15-8 NOT ALL 1'S ON DETECTION OF ILLEGAL
                                        ;*FORMAT CODE #7
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 07
                                        ;*WMC1 MICRO ERROR 31
                                        ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
                                        ;*M8959
                                        ;*WMC ERROR STATUS DOES NOT INDICATE 'ROM ERROR' FROM ILLEGAL FORMAT CODE #7
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 07
                                        ;*WMC1 MICRO ERROR 32
                                        ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
                                        ;*M8959
                                        ;*WMC ERROR STATUS DOES NOT INDICATE 'ERROR' FROM ILLEGAL FORMAT CODE #7
                                        ;*ACTUAL - NNNN
45F5                                    $
 (1)                                    ; *****************************************************************************
                                        TEST7:  TESTX   @7
45F5                                                    MVI     A,@7            ;DEFINE THE TEST NUMBER
 (1)  45F5  3E  07          7.0                         CALL    TSET            ;SETUP THE TEST
 (1)  45F7  CD  03  28     18.0
                                        ;%WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
                                        ;&M8959

                                        ;RESTART THE WMC PROGRAM
45FA  3E  01               7.0          TST7L:  MVI     A,W.RST
45FC  D3  D3              10.0                  OUT     WMCCTL          ;ISSUE WMC RESTART
45FE  AF                   4.0                  XRA     A
45FF  D3  D3              10.0                  OUT     WMCCTL          ;CLEAR THE RESTART BIT

                                        ;INIT THE 'ECODE' COUNTER TO 0

4601                                            CLRECT
 (1)  4601  AF              4.0                         XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4602  D3  D6         10.0                         OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4604  D3  D6         10.0                         OUT     ERRCNT          ;CLEAR BITS 15-8
```

```
2046
2047                                    ;LOAD INVALID FORMAT CODE #7 IN WMC DATA CONTROL WORD
2048
2049  4606  3E 70            7.0       MVI     A,@160          ;GET CODE #7
2050  4608  D3 D0           10.0       OUT     DATACTL         ;LOAD IT IN WMC DATA CONTROL WORD
2051                                    ;SET WMC CONTROL TO 'WMC ENABLE' SO THE ERROR CAN BE DETECTED
2052
2053  460A  3E 80            7.0  T7F7A: MVI     A,W.ENAB        ;ENABLE WMC
2054  460C  D3 D3           10.0       OUT     WMCCTL
2055  460E  3E 0A            7.0       MVI     A,10            ;GET A DELAY COUNT
2056  4610  3D               4.0  T7W:   DCR     A
2057  4611  C2 10 46        10.0       JNZ     T7W             ;WAIT TILL COUNT TO 0
2058  4614  DB D0           10.0       IN      WMCSTA          ;GET WMC STATUS INFO
2059  4616  E6 40            7.0       ANI     W.DONN          ;CHECK TO SEE IF WMC PROGRAM IS READY
2060  4618  CA 20 46        10.0       JZ      T7F7AC          ;JUMP OVER IF WMC IS DONE
2061  461B                            ERR     T7F7L,T7F7AC
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  461B  CD 09 28        18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0016                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  461E  16                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  461F  00                       .BYTE
 (1)  4620  CD 15 28        18.0  T7F7AC::      CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4623  DA FA 45        10.0                JC      T7F7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2062                                    ;>WMC NOT READY - WMC GROSS ERROR! POSSIBLE WMC SELF-DIAGNOSIS FAILURE.
2063
2064  4626  DB D6           10.0  T7F7B: IN      ERRCNT          ;GET BITS 7-0
2065  4628  47               4.0       MOV     B,A             ;TEMP SAVE
2066  4629                            ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  4629  D3 94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  462B  7F               4.0                MOV     A,A             ;RETRY LINK
2067  462C  DB D6           10.0       IN      ERRCNT          ;GET BITS 15-8 OF 'ECODE' COUNTER
2068  462E  4F               4.0       MOV     C,A             ;TEMP SAVE BITS
2069  462F  78               4.0       MOV     A,B             ;GET 7-0 AGAIN
2070  4630  FE FD            7.0       CPI     @375            ;SET TO MINUS 3??
2071  4632  CA 3A 46        10.0       JZ      T7F7BC          ;JUMP IF CORRECT
2072  4635                            ERRA    T7F7L,T7F7BC
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4635  CD 0F 28        18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0017                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4638  17                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4639  00                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  463A  CD 15 28        18.0  T7F7BC::      CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  463D  DA FA 45        10.0                JC      T7F7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2073                                    ;>WMC 'ECODE' COUNT NOT -3 (3/5) AFTER ILLEGAL FORMAT CODE #7 USED
2074
2075  4640  79               4.0  T7F7C: MOV     A,C             ;GET BITS 15-8 OF 'ECODE'
2076  4641                            ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  4641  D3 94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4643  7F               4.0                MOV     A,A             ;RETRY LINK
2077  4644  FE FF            7.0       CPI     @377            ;SHOULD BE ALL 1'S
2078  4646  CA 4E 46        10.0       JZ      T7F7CC          ;JUMP IF CORRECT
```

```
2080  4649                                        ERRA    TST7L,TST7CC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4649  CD  0F  28        18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0018                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  464C  18                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  464D  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  464E  CD  15  28        18.0        TST7CC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4651  DA  FA  45        10.0                JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2081                                        ;>WMC 'ECODE' COUNTER BITS 15-8 NOT ALL 1'S ON DETECTION OF ILLEGAL
2082                                        ;>FORMAT CODE #7
2083
2084                                        ;CHECK WMC ERROR STATUS WORD FOR 'ERROR' INDICATION FROM THE ILLEGAL
2085                                        ;FORMAT CODE
2086
2087  4654  DB  DA                10.0  TST7D:  IN      WMCERR          ;GET THE ERROR STATUS
2088  4656  E6  10                 7.0          ANI     W.ROME          ;SEE IF 'ERROR' IS SET
2089  4658  C2  60  46           10.0          JNZ     TST7DC          ;JUMP IF IT IS SET
2090  465B                                    ERRA    TST7L,TST7DC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  465B  CD  0F  28        18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0019                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  465E  19                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  465F  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4660  CD  15  28        18.0        TST7DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4663  DA  FA  45        10.0                JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2091                                        ;>WMC ERROR STATUS DOES NOT INDICATE 'ROM ERROR' FROM ILLEGAL FORMAT CODE #7
2092
2093  4666  DB  DA                10.0  TST7E:  IN      WMCERR          ;GET THE ERROR STATUS
2094  4668  E6  20                 7.0          ANI     W.ERR           ;SEE IF 'ERROR' IS SET
2095  466A  C2  72  46           10.0          JNZ     TST7EC          ;JUMP IF IT IS SET
2096  466D                                    ERRA    TST7L,TST7EC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  466D  CD  0F  28        18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        001A                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4670  1A                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4671  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4672  CD  15  28        18.0        TST7EC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4675  DA  FA  45        10.0                JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2097                                        ;>WMC ERROR STATUS DOES NOT INDICATE 'ERROR' FROM ILLEGAL FORMAT CODE #7
2098  4678                                      ENDTST  TST7L
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4678                                    REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4678  CD  06  28        18.0                CALL    REQST
 (2)  467B  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (2)  467C  00  00                                .WORD                   ;SYSTEM '*' COUNT
 (2)  467E  00  00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4680  00                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4681  07                                    .BYTE   7               ;REQUEST CODE
 (1)  4682  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4685  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  4686  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  4689  F2  FA  45        10.0          JP      TST7L           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - WMC SYSTEM CLOCK TEST
                               ST
                               ;*********************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*SYSTEM CLOCK TEST
                               SM
                               ;*********************************************************************
                               ;*MODULE(S) UNDER TEST
                               ;*--------------------
                               ;*M8959,M8960
                               SD
                               ;*********************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS TEST VERIFYS ALL SYSTEM CLOCKS FOR THE WMC
                               ;*BY UTALIZING THE ILLEGAL FORMAT CODE #7 JUST TESTED
                               ;*
                               ;*FIRST THE -20% CLOCK IS APPLIED TO THE WMC - THEN
                               ;*A RESTART AND ILLEGAL FORMAT 7 OPERATION ARE PERFORMED
                               ;*AND THE WMCERR WORD IS MONITORED FOR THE EXPECTED
                               ;*ERROR - FAILURE OF IHE EXPECTED ERROR CONDITION
                               ;*TO SET INDICATES THE WMC IS NOT RECEIVENG THE SELECTED CLOCK.
                               ;*THE PROCEDURE IS THEN REPEATED FOR THE +10%, +20%, AND NORMAL CLOCKS.
                               SP
                               ;*********************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*    SET SYSTEM CCLOCK TO -20%
                               ;*       BGNDO
                               ;*          ISSUE A WMC RESTART
                               ;*          SET FORMAT 7 - ILLEGAL
                               ;*          SET 'WMC ENABLE' TO START PROCESSING
                               ;*          SMALL DELAY
                               ;*          IF ROM ERROR AND ERROR SET
                               ;*             THEN - CONTINUE
                               ;*             ELSE - ERROR
                               ;*          ENDIF
                               ;*          ISSUE A 'WMC RESTART' TO CLEANUP
                               ;*          IF ROM ERROR AND ERROR CLEAR
                               ;*             THEN - CONTINUE
                               ;*             ELSE - ERROR
                               ;*          ENDIF
                               ;*          CHANGE SYSTEM CLOCKS AND DO AGAIN UNTIL CLOCK=NORMAL
                               ;*       ENDDO
                               ;*ENDTST
                               SE
                               ;*********************************************************************
                               ;*ERRORS
                               ;*------
                               ;*WMC1 MICRO TEST 10
                               ;*WMC1 MICRO ERROR
                               ;*WMC1-SYSTEN CLOCK TEST
```

```
                                        ;*M8959, M8960
                                        ;*NO ERROR SET AFTER ILLEGAL FORMAT CODE 7
                                        ;*ACTUAL = NNNN
                                        ;*ACTUAL DATA IS THE CLOCK CONTROLE WORD AT THE TIME OF FAILURE
                                        S
                                        ; ************************************************************************
                                        TEST10: TESTX   @10
                                                        MVI     A,@10           ;DEFINF THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%WMC1 - SYSTEM CLOCK TEST
                                        ;&M8959, M8960

                                        TST10R: MVI     A,@03           ;SET THE SYSTEM CLOCK TO -20%
                                                STA     SPEED           ;SAVE THE CLOCK FOR POSSIBLE ERROR REPORT
                                        TST10L: OUT     CLKCTL

                                                MVI     A,W.RST         ;ISSUE WMC RESET
                                                OUT     WMCCTL          ;
                                                XRA     A               ;CLEAR THE RESTART BIT
                                                OUT     WMCCTL          ;TO FINISH THE RESET

                                                MVI     A,@160          ;LOAD FORMAT CODE 7 - ILLEGAL
                                                OUT     DATACTL         ;
                                                MVI     A,W.ENAB        ;ENABLE WMC
                                                OUT     WMCCTL          ;
                                                                        ;

                                        T10W:   DCR     A               ;DELAY FOR OPERATION TO COMPLEAT
                                                JNZ     T10W            ;

                                                IN      WMCERR          ;CHECKL FOR EXPECTED ERROR
                                                ANI     W.ROME+W.ERR    ;
                                                JNZ     TST10A          ;CORRECT - CONTINUE

                                                LDA     SPEED           ;GET SYSTEM CLOCK FOR ERROR REPORT
                                                ROUT    ADATA           ;
                                                        OUT     ADATA           ;WRITE AC INTO ADATA
                                                        MOV     A,A             ;RETRY LINK

                                                ERRA    TST10L,TST10A   ;
                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
                                                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                TST10A::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      TST10L          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>NO ERROR SET AFTER ILLEGAL FORMAT CODE 7
                                        ;<ACTUAL DATA IS THE CLOCK CONTROLE WORD AT THE TIME OF FAILURE

                                                LDA     SPEED           ;GET THE CLOCK CONTROLE WORD
                                                ANA     A               ;SET CONDITION BITS
```

```
2181  46C9  CA  D3  46   10.0          JZ      TST10B          ;ALL CLOCKS TESTED - GET OUT
2182  46CC  3D                 4.0     DCR     A               ;NO - GO DO IT AGAIN WITH NEXT CLOCK
2183  46CD  32  EA  46   13.0          STA     SPEED           ;SAVE THE CLOCK CONTROLE WORD
2184  46D0  C3  96  46   10.0          JMP     TST10L          ;
2185
2186  46D3                        TST10B: ENDTST  TST10R
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46D3                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46D3  CD  06  28'   18.0                      CALL    REQST
 (2)  46D6  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  46D7  00  00                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  46D9  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46DB  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46DC  07                                      .BYTE   7               ;REQUEST CODE
 (1)  46DD  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  46E0  3D                 4.0     DCR     A               ;DOWNCOUNT
 (1)  46E1  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  46E4  F2  91  46   10.0          JP      TST10R          ;DO TEST UNTIL TILL = 0
2187  46E7  C3  18  28   10.0          JMP     TSTEND          ;END OF THE TEST
2188
2189
2190
2191  46EA  00                    SPEED:  .BYTE   0               ;STORAGE FOR CLOCK CONTROLE WORD
2192        0000                  .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLKCTL= 00FC           CLOCK   4F26           CMC0H  = 0099          CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CRCWRD= 0018           CSAVE   4F9D          CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F
CTSTL  = C08E          CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002
D      =%0002          DATACT= 00D0           DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045          DONINT= 0010          DSAVE   4F9E
DSE    = 0006          D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          E      =%0003          ECCBAD= 0042          ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095
EOTCLR= 0003           ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90          ERRCNT= 00D6
ESAVE   4F9F           E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25
FWDTST= 0061           GCRID  = 0089          GCRSET= 0002          GOODTM= 0092
H      =%0004          HLSAVE  4FA0           IE     = 0008          INTSTA= 00E0
ITERA   4F9A           I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F          KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D          KEY11  = 006E          KEY12  = 006F
KEY13  = 005C          KEY14  = 005D          KEY15  = 005E          KEY16  = 005F
KEY17  = 003C          KEY18  = 003D          KEY19  = 003E          KEY2   = 0079
KEY20  = 003F          KEY3   = 007A          KEY4   = 007B          KEY5   = 0074
KEY6   = 0075          KEY7   = 0076          KEY8   = 0077          KEY9   = 006C
KINTA  = 006F          KLDAD  = 003D          KN0    = 003C          KN1    = 005C
KN2    = 005D          KN3    = 005E          KN4    = 006C          KN5    = 006D
KN6    = 006E          KN7    = 0074          KN8    = 0075          KN9    = 0076
KU2    = 0079          KU3    = 007A          KU8    = 0077          L      =%0005
LBLANK= 000F           LCE    = 0008          LCH    = 000C          LCL    = 000D
```

```
LCP    = 000E          LC0    = 0000          LC1    = 0001          LC2    = 0002
LC3    = 0003          LC4    = 0004          LC5    = 0005          LC6    = 0006
LC7    = 0007          LC8    = 0008          LC9    = 0009          LDLEDA= 00CA
LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD           LDLEDE= 00CE
LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C          LKKEY  = 0049
LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052           LKLWPP= 004F
LKMOD7= 0046           LKOPR  = 0046          LPFLG   4F94           LPNUM   4F92
M      =%0006          MBSEL  = 00E0          MB.A   = 0008          MB.B   = 0004
MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000          MSE    = 0008
MSGN   = 001B          MTACLR= 0000           MT.ARA= 0020           MT.CPE= 0080
MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008           MT.LWR= 0004
MT.MOT= 0002           MT.NWT= 0080           MT.PEC= 0040           MT.PSB= 0004
MT.PS0= 0001           MT.PS1= 0002           MT.REV= 0020           MT.WRT= 0010
MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020           M.CONT= 0080
M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008          M.FAIL= 0008
M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020          M.ONLI= 0001
M.PE   = 0040          M.PORT= 0080           M.RDEN= 0002           M.RDPE= 0008
M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040          M.UNIT= 0007
M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080           M5.5   = 0001
M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088           OKAY   = 00FF
OPRRAM= 4300           OPSTRT= 0058           OPVER  = 0040          PADCNT= 00D5
PADCRC= 0080           PDIAG  = 0048          PEID   = 008A          PENAB  = 004C
PESET  = 0001          PL     = 0081          PRDD   = 004C          PRENF  = 009C
PS     = 0082          PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001
P.BCTC= 0040           P.CMDP= 0020           P.INTE= 0080           P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004           P.RPST= 0002
P.RPOE= 0020           P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010
P.SING= 0080           P.STAT= 0002           P.STPE= 0080           P.TACH= 0008
P.TUPR= 0010           P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010           P.WP0E= 0008           P.WP1E= 0004           P.WP2E= 0008
P.WP3E= 0004           P.5VOK= 0002           QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047           RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          REND   = 0014
REQST  = 2806          RESCHR= 00D1           REVTST= 0064           REWIND= 0004
RFIFOL= 0008           RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001
RILL   = 0012          RINST  = 000C          RMCTST= 0008           RMK2   = 0013
RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001
RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000
RPF1   = 009D          RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016
RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A
RTIER  = 0030          RTM    = 0005          RUNKI  = 0009          RUPTST= 005E
RWDUNL= 0005           R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040
R.DON  = 0002          R.DRDY= 0010           R.END  = 0010          R.ILL  = 0004
R.JVOK= 0004           R.MK2  = 0008          R.PLOD= 0008           R.PLOO= 0010
R.PLO1= 0020           R.POST= 0020           R.STNM= 0002           R.STOP= 0004
R.STPC= 0001           R.TBJN= 0080           R.TSTD= 0040           R.VOK  = 0080
R00H   = 0081          R00L   = 0080          R01H   = 0083          R01L   = 0082
R02H   = 0085          R02L   = 0084          R03H   = 0087          R03L   = 0086
R04H   = 0089          R04L   = 0088          R05H   = 008B          R05L   = 008A
R06H   = 008D          R06L   = 008C          R07H   = 008F          R07L   = 008E
R10H   = 0091          R10L   = 0090          R11H   = 0093          R11L   = 0092
R12H   = 0095          R12L   = 0094          R13H   = 0097          R13L   = 0096
```

```
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SPEED   46EA
SSCLK = 0040          SSTEP = 0005          STACK = 4FFF          STATRM= 4F20
STPCT   4F20          STRSP = 5000          TADR00= 0080          TADR01= 0081
TADR02= 0082          TADR03= 0083          TADR04= 0084          TADR05= 0085
TADR06= 0086          TADR07= 0087          TADR10= 0088          TADR11= 0089
TADR12= 008A          TADR13= 008B          TAMT  = 0044          TASEL = 0080
TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004
TC.REV= 0020          TC.WRT= 0010          TEMP    4F99          TEST1   4300
TEST10  468C          TEST2   4386          TEST3   43FE          TEST4   4443
TEST5   44E0          TEST6   4568          TEST7   45F5          TMF   = 0099
TMRDY = 0040          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TST1AC  433C  G       TST1B   4342          TST1BC  4351  G
TST1C   4357          TST1CC  436C  G       TST1L   4325          TST1W   4359
TST10A  468F  G       TST10B  46D3          TST10L  4696          TST10R  4691
TST2AC  43A4  G       TST2BC  43BC  G       TST2CC  43D1  G       TST2D   43D7
TST2DC  43E4  G       TST2L   438B          TST3A   4407          TST3AC  4413  G
TST3B   441D          TST3BC  4429  G       TST3L   4403          TST4AC  4479  G
TST4B   447F          TST4BC  448D  G       TST4C   4493          TST4CA  44B2  G
TST4D   44B8          TST4DC  44C6  G       TST4L   445A          TST4ST  4448
TST5AC  4504  G       TST5B   450A          TST5BC  4518  G       TST5C   451E
TST5CA  453D  G       TST5D   4543          TST5DC  4551  G       TST5L   44E5
TST6AC  458F  G       TST6B   4595          TST6BC  45A3  G       TST6C   45A9
TST6CA  45C8  G       TST6D   45CE          TST6DC  45DB  G       TST6L   4570
TST7A   460A          TST7AC  4620  G       TST7B   4626          TST7BC  463A  G
TST7C   4640          TST7CC  464E  G       TST7D   4654          TST7DC  4660  G
TST7E   4666          TST7EC  4672  G       TST7L   45FA          TUSEL0= 00D1
TUSEL1= 00D2          TU78  = 0010          T.ATH0= 0001          T.ATH1= 0002
T.BOT = 0004          T.EOT = 0002          T.FPT = 0001          T.NTHR= 0004
T.ONL = 0020          T.PES = 0008          T.PSBJ= 0020          T.PS0J= 0008
T.PS1J= 0010          T.RDY = 0080          T.RDYO= 0040          T.RWD = 0010
T.SCLK= 0002          T10W    46A9          T4W     4456          T7W     4610
UIBG  = 00A1          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAC  0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002
X.ROME= 0001          X.WCLK= 0001          Y     =%000B          .     = 46EB
```

ERRORS DETECTED: 0

*WMC1.A78/PTP,WMC1=NLIST,PARAM,MACRO,LIST,WMC1
RUN-TIME: 4 6 0 SECONDS
CORE USED: 10K

```
                                .TITLE  MBD1 - MASSBUS DATA TEST PART #1
                                .SBTTL  TEST 1 - WRITE DATA TRANSFER USING THE 'DDR' REGISTERS - WMC IN 'RIGHT' MODE
                                ;ID     MBD1-MASSBUS DATA CONTROLLER
4300                            ST
                                ; ********************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*DATA TRANSFER FROM HOST TO TM78 IN 'RIGHT' MODE
4300                            SM
                                ; ********************************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8956, M8957, M8959
4300                            SD
                                ; ********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST REQUIRES THE HOST CPU TO GENERATE THE 'MBCPAR' DATA PATTERN
                                ;*TO SEND OVER THE MASSBUS ON A WRITE GCR COMMAND (FUNCTION CODE 63). IF
                                ;*THE HOST CPU IS NOT CAPABLE OF AN 18 BIT DATA TRANSFER OVER THE
                                ;*MASSBUS, THEN THE TEST EXPECTS A FUNCTION CODE OF 65 TO BE RETURNED!
                                ;*
                                ;*THE HOST WILL BE REQUESTED TO TRANSFER 100 BYTES (18 BIT) OF DATA USING THE
                                ;*10 DUMP MODE PACKING FORMAT (5 BYTES PER 10 CPU WORD). THE BYTE COUNT
                                ;*REQUESTED WILL APPEAR IN THE MASSBUS REGISTER 05 (BYTE COUNT REGISTER).
                                ;*
                                ;*MASSBUS REGISTER 02 WILL CONTAIN THE RECORD COUNT (SET TO 1) AND THE
                                ;*PACKING MODE TO USE (CODE =2 FOR 10 DUMP) - NO UNIT IS NEEDED FOR THE
                                ;*TEST.
                                ;*
                                ;*THIS TEST WILL ISSUE 40 'SCLK' CYCLES TO TRANSFER THE DATA FROM THE
                                ;*HOST CPU. DATA IS CHECKED AFTER EACH CLOCK CYCLE AND WILL CAUSE AN
                                ;*ERROR REQUEST TO THE HOST IF FOUND TO BE INCORRECT!
                                ;*
                                ;*DETECTION OF A MASSBUS DATA PARITY ERROR OR 'OTHER' ERROR STATUS IN
                                ;*THE WMC ERROR STATUS WORD (100332) ALSO CAUSES AN ERROR REPORT.
                                ;*
                                ;*DATA ERRORS OR STATUS ERRORS WILL TERMINATE THE DATA TRANSFER IF HOST
                                ;*CPU HAS REQUESTED THE MICROCOMPUTER TO 'LOOP-ON-ERROR'. INTERNAL STATUS
                                ;*ERRORS WILL BE CLEARED IF REPORTED TO BE SET.
                                ;*
                                ;*MASSBUS DATA TEST - USE HOST CPU TO CHECK MBD MODULE OK FOR WRITE
                                ;*AND READ DATA TRANSFERS.
                                ;*WRITE XFR TEST
4300                            SP
                                ; ********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   INIT TEST - TESTX MACRO
                                ;*   ISSUE WMC RESTART TO INIT WRITE PATH
                                ;*   INIT ECODE COUNTER TO 0 AND SELECT WMC CONTROL TO 'WRITE'
                                ;*   SET DATA BUS CONTROL TO 0 + WAIT SMALL AMOUNT OF TIME
                                ;*   IF DBUS STATUS STILL-0
```

```
1371                                    ;*  :  THEN-CONTINUE
1372                                    ;*  :  ELSE-REPORT ERROR - INCORRECT DBUS
1373                                    ;*  :       STATUS - MAYBE CABLE FAILURE
1374                                    ;*  ENDIF
1375                                    ;*  INIT DDR REGS TO 'IN' + REQUEST HOST CPU FOR A 'WRITE' GCR DATA XFR
1376                                    ;*  CONTINUE WITH HOST RESPONSE - CODE 63 OR 67 (IF HOST NOT CAPABLE OF 18 BIT XFR)
1377                                    ;*  DATA SENT BY HOST IS 'MBDPAR' IN 10 DUMP MODE FORMAT
1378                                    ;*  INIT 'SCLK' COUNT MASSBUS CONTROL TO 'OCC'-'RUN'-'MBRDEN'
1379                                    ;*  SET 'SCLK' + WAIT FOR HOST RESPONSE WITH 'WCLK'
1380                                    ;*  IF 'WCLK' RESPONSE BY HOST
1381                                    ;*  :  THEN-REPORT NO HOST WCLK' RESPONSE-OPI ERROR
1382                                    ;*  :  ELSE-CONTINUE
1383                                    ;*  ENDIF
1384                                    ;*  FINISH 1ST 18 BIT 'SCLK' CYCLE BY CLEARING 'SCLK' BIT FROM DBUS CONTROL
1385                                    ;*  IF 'SCLK' GO AWAY
1386                                    ;*  :  THEN-REPORT INCORRECT DBUS STATUS
1387                                    ;*  :  ELSE-CONTINUE
1388                                    ;*  ENDIF
1389                                    ;*  GET DATA IN DDR REGS 'A', 'B', + 'C' + CHECK IF DATA OK
1390                                    ;*  IF DATA RECEIVED OK
1391                                    ;*  :  THEN-REPORT READ FAILURE OF DATA FROM HOST CPU
1392                                    ;*  :  ELSE-CONTINUE
1393                                    ;*  ENDIF
1394                                    ;*  BGNDO
1395                                    ;*  :  DO UNTIL ALL DATA CLOCKED FROM HOST AND CHECKED
1396                                    ;*  :  CLOCK NEXT 18 BITS ACROSS MASSBUS
1397                                    ;*  :  IF DATA RECEIVED OK
1398                                    ;*  :  :  THEN-CONTINUE
1399                                    ;*  :  :  ELSE-REPORT DATA XFR ERROF + 'SCLK' COUNT
1400                                    ;*  :  ENDIF
1401                                    ;*  ENDDO
1402                                    ;*  DROP MASSBUS SIGNALS 'OCC' + 'RUN' TO RELEASE THE HOST CPU
1403                                    ;*ENDTST
1404  4300                              SE
 (1)                                    ;*************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1405                                    ;*MBD1 MICRO TEST 01
1406                                    ;*MBD1 MICRO ERROR 01
1407                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-'DDR' IN 'RIGHT' MODE
1408                                    ;*M8956, M8957, M8959
1409                                    ;*DATA BUS CONTROL WORD INITED TO 0 AND WAS READ NON-0
1410                                    ;*POSSIBLE MASSBUS CABLE FAILURE
1411                                    ;*ACTUAL = NNNN
1412                                    ;*EXPECTED = NNNN
1413                                    ;*
1414                                    ;*MBD1 MICRO TEST 01
1415                                    ;*MBD1 MICRO ERROR 02
1416                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-'DDR' IN 'RIGHT' MODE
1417                                    ;*M8956, M8957, M8959
1418                                    ;*OPI ERROR - HOST CPU FAILED TO SET 'RUN' ON A WRITE COMMAND (6X CODE)
1419                                    ;*TEST ABORTED!
1420                                    ;*
1421                                    ;*MBD1 MICRO TEST 01
```

```
1422                            ;*MBD1 MICRO ERROR 03
1423                            ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1424                            ;*M8956, M8957, M8959
1425                            ;*HOST CPU FAILED TO RESPOND TO "SCLK" COMMAND WITH A "WCLK" ON THE
1426                            ;*FIRST "SCLK" ISSUED - DATA FAILED TO TRANSFER OVER MASSBUS
1427                            ;*ACTUAL = NNNN
1428                            ;*EXPECTED - NNNN
1429                            ;*TEST ABORTED!
1430                            ;*
1431                            ;*MBD1 MICRO TEST 01
1432                            ;*MBD1 MICRO ERROR 04
1433                            ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1434                            ;*M8956, M8957, M8959
1435                            ;*EXPECTED CBUSSTA BYTE NOT = ACTUAL
1436                            ;*"SCLK" FAILED TO CLEAR AFTER IT WAS SET.
1437                            ;*ACTUAL = NNNN
1438                            ;*EXPECTED = NNNN
1439                            ;*TEST ABORTED.
1440                            ;*
1441                            ;*MBD1 MICRO TEST 01
1442                            ;*MBD1 MICRO ERROR 05
1443                            ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1444                            ;*M8956, M8957, M8959
1445                            ;*MBD BITS 7-0 COMPARE ERROR ON THIS "SCLK" XFR
1446                            ;*ACTUAL = NNNN
1447                            ;*EXPECTED = NNNN
1448                            ;*BYTE/SCLK COUNT NUMBER = LLL
1449                            ;*EXPECTED 18 BITS = E EEEEEE
1450                            ;*ACTUAL 18 BITS   = A AAAAAA
1451                            ;*TEST ABORTED!
1452                            ;*
1453                            ;*MBD1 MICRO TEST 01
1454                            ;*MBD1 MICRO ERROR 06
1455                            ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1456                            ;*M8956, M8957, M8959
1457                            ;*MBD BITS 15-8 COMPARE ERROR ON THIS "SCLK" XFR
1458                            ;*ACTUAL - NNNN
1459                            ;*EXPECTED - NNNN
1460                            ;*BYTE/SCLK COUNT NUMBER = LLL
1461                            ;*EXPECTED 18 BITS = E EEEEEE
1462                            ;*ACTUAL 18 BITS   = A AAAAAA
1463                            ;*
1464                            ;*MBD1 MICRO TEST 01
1465                            ;*MBD1 MICRO ERROR 07
1466                            ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1467                            ;*M8956, M8957, M8959
1468                            ;*MBD BITS 17-16 AND "P" INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
1469                            ;*OF WMC ERROR WORD - 100332).
1470                            ;*ACTUAL = NNNN
1471                            ;*EXPECTED = NNNN
1472                            ;*BYTE/SCLK COUNT NUMBER = LLL
1473                            ;*EXPECTED 18 BITS = E EEEEEE
1474                            ;*ACTUAL 18 BITS   - A AAAAAA
1475                            ;*
```

```
                                        ;*MBD1 MICRO TEST 01
                                        ;*MBD1 MICRO ERROR 10
                                        ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-'DDR' IN 'RIGHT' MODE
                                        ;*M8956, M8957, M8959
                                        ;*MASSBUS PARITY ERROR DETECTED BY THE TM78
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*TEST ABORTED!
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*EXPECTED 18 BITS = E EEEEEE
                                        ;*ACTUAL 18 BITS   = A AAAAAA
                                        $
 (1)                                    ; ************************************************************************************

                                        TEST1:  TESTX   @1
 (1) 4300  3E  01          7.0                   MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%MBD1-DATA WRITE TO TM78 TEST USING DIAG. DATA REG.-'DDR' IN 'RIGHT' MODE
                                        ;&M8956, M8957, M8959
                                        ;DISABLE THE MICROCOMPUTER INTERRUPT SYSTEM

4305  AF                  4.0   TST1L:  XRA     A               ;SET THE SYSTEM CLOCK TO 'NORMAL'
4306  D3  F0            10.0           OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD

4308  3E  01              7.0           MVI     A,W.RST         ;RESTART THE WMC
430A  D3  D3             10.0           OUT     WMCCTL
430C  AF                  4.0           XRA     A               ;FINISH THE RESTART CYCLE
430D  D3  D3             10.0           OUT     WMCCTL          ;DROP THE RST BIT IN WMC CONTROL WORD
430F                                    INICNT                  ;INIT THE COUNTERS
 (1) 430F  3E  30          7.0                           MVI     A,@060          ;INIT THE
 (1) 4311  D3  D7         10.0                           OUT     CNTCTL          ;BYTE COUNTER
 (1) 4313  3E  70          7.0                           MVI     A,@160          ;INIT THE
 (1) 4315  D3  D7         10.0                           OUT     CNTCTL          ;PAD COUNTER
 (1) 4317  3E  B0          7.0                           MVI     A,@260          ;INIT THE
 (1) 4319  D3  D7         10.0                           OUT     CNTCTL          ;ERROR COUNTER
431B                                    CLRECT
 (1) 431B  AF              4.0                           XRA     A               ;CLEAR THE ACCUMULATOR
 (1) 431C  D3  D6         10.0                           OUT     ERRCNT          ;CLEAR BITS 7-0
 (1) 431E  D3  D6         10.0                           OUT     ERRCNT          ;CLEAR BITS 15-8
4320  3E  08              7.0           MVI     A,W.WRITE       ;ENABLE ONLY 'WRITE' IN WMC CNTRL WORD
4322  D3  D3             10.0           OUT     WMCCTL          ;SET WRITE STATUS

4324  3E  00              7.0           MVI     A,0             ;AGAIN FOR THE LOOP
4326                                    ROUT    EDATA           ;SAVE 'EDATA' ON STATUS CHECK
 (1) 4326  D3  95         10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1) 4328  7F              4.0                   MOV     A,A             ;RETRY LINK
4329  D3  C0             10.0           OUT     DBUSCTL         ;CLEAR THE DBUS CNTRL WORD
432B  00                  4.0           NOP                     ;SMALL DELAY
432C  DB  C0             10.0           IN      DBUSSTA         ;GET THE DBUS STATUS
432E  A7                  4.0           ANA     A               ;SET THE CONDITION BITS
432F                                    ROUT    ADATA           ;SAVE THE 'ADATA' IN CASE OF ERROR
 (1) 432F  D3  94         10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4331  7F              4.0                   MOV     A,A             ;RETRY LINK
4332  CA  3A  43         10.0           JZ      TST1AC          ;JUMP OVER IF NO ERROR - SHOULD BE  0
```

```
1514  4335                                          ERRB    TST1L,TST1AC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4335  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0001                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4338  01                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4339  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  433A  CD  15  28        18.0              TST1AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  433D  DA  05  43        10.0                          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1515                                            ;>DATA BUS CONTROL WORD INITED TO 0 AND WAS READ NON-0
1516                                            ;>POSSIBLE MASSBUS CABLE FAILURE
1517                                            ;STATUS WAS OK - PROCEED TO SETUP THE DDR REGISTERS (CLEAR THEM)
1518
1519  4340  3E  9B           7.0  TST1B:  MVI     A,@233          ;SET DDR CNTRL WORD TO ''IN''
1520  4342  D3  DB          10.0         OUT     DDRCTL
1521                                            ;MASSBUS REGISTERS 02 (RECORD COUNT, FMT TYPE, SKIP CNT) AND REGISTER
1522                                            ;05 (BYTE COUNT REG) WILL BE SET UP ON THE DIAGNOSTIC MONITOR CALL.
1523
1524                                            ;REQUEST THE DIAG. MON. TO SIGNAL THE HOST CPU FOR A WRITE GCR DATA
1525                                            ;TRANSFER USING ''MBDPAR'' DATA PATTERN (2ND ARG IN CALL). BYTE COUNT
1526                                            ;IS 100. (40 SCLK * 2.5 BYTES PER WORD IN 10 DUMP MODE). REG 02 GETS
1527                                            ;LOADED TO 1 RECORD @ 10 DUMP FORMAT.
1528
1529  4344                                          REQ     2,1,40,@30004
 (1)  4344  CD  06  28        18.0                          CALL    REQST
 (1)  4347  01                                            .BYTE   1               ;DATA PATTERN NUMBER
 (1)  4348  00  28                                        .WORD   40              ;SYSTEM ''40'' COUNT
 (1)  434A  30  04                                        .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  434C  00                                            .BYTE                   ;DATA COMPARE FLAG IF -1
 (1)  434D  02                                            .BYTE   2               ;REQUEST CODE
1530
1531                                            ;INIT THE SCLK COUNTER TO 0 ... TEST WILL STOP WHEN COUNT = 41.
1532
1533  434E  3E  01           7.0                  MVI     A,1
1534  4350  32  53  46      13.0                  STA     CLKCNT          ;INIT THE CLOCK COUNTER
1535
1536                                            ;SET THE DBUS CNTRL WORD TO ''OCC'', AND ''SCLK OUT''
1537
1538  4353  3E  22           7.0                  MVI     A,M.OCC!T.SCLK
1539  4355  D3  C0           10.0                 OUT     DBUSCTL         ;SET THE STATUS - NOW HAVE MASSBUS
1540                                            ;NOW WAIT FOR HOST CPU TO ASSERT ''RUN'' ON THE MASSBUS...COULD TAKE UP TO
1541                                            ;10 MILLISEC (MASSBUS SPEC.)
1542
1543  4357  0E  14           7.0                  MVI     C,20            ;LOOK FOR 100 M-SEC.
1544  4359  DB  C0           10.0  T1W:     IN      DBUSSTA         ;GET DBUS STATUS - CHECK FOR ''RUN''
1545  435B  E6  04           7.0                  ANI     M.RUN           ;SAVE ONLY ''RUN'' BIT
1546  435D  C2  7B  43       10.0                 JNZ     TST1C           ;JUMP IF RUN IS UP - OK TO PROCEED
1547  4360  3E  FF           7.0  T1WL:    MVI     A,@377          ;GET A DELAY TIME
1548  4362  3D               4.0  T1WL1:   DCR     A               ;DECREMENT
1549  4363  C2  62  43       10.0                 JNZ     T1WL1           ;STAY IN LOOP TILL =0
1550
1551                                            ;DOWNCOUNT GROSS TIMER COUNT
1552
1553  4366  0D               4.0                  DCR     C
1554  4367  C2  59  43       10.0                 JNZ     T1W             ;THEN CHECK IF ''RUN'' IS UP
```

```
                                        CALL    QUIT
                                        ERR     TSTEND,T1WLC
                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
                                                CALL    ERLP            ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE
                                        T1WLC:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
                                                JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>OPI ERROR - HOST CPU FAILED TO SET 'RUN' ON A WRITE COMMAND (6X CODE)
                                ;<TEST ABORTED!
                                        JMP     TSTEND          ;FATAL ERROR! - QUIT NOW
                                ;START THE CLOCK CYCLE (SCLK) BY SETTING THE SCLK BIT IN DBUS CNTRL
                                ;WORD

                        TST1C:  MVI     A,M.OCC!T.SCLK!M.SCLK   ;'OCC' + 'SCLK OUT' + 'SCLK'
                                OUT     DBUSCTL         ;SET SCLK BIT
                                ORI     M.RUN!M.WCLK    ;ADD IN OTHER EXPECTED BITS
                                ROUT    EDATA           ;SAVE FOR ERROR REPORTING
                                        OUT     EDATA           ;WRITE AC INTO EDATA
                                        MOV     A,A             ;RETRY LINK
                                NOP
                                IN      DBUSSTA         ;GET THE DBUS STATUS - CHECK FOR WCLK
                                ROUT    ADATA           ;SAVE ACTUAL DATA READ
                                        OUT     ADATA           ;WRITE AC INTO ADATA
                                        MOV     A,A             ;RETRY LINK
                                ANI     M.WCLK          ;SAVE ONLY 'WCLK' BIT
                                JNZ     TST1CB          ;JUMP OVER IF SET - NO ERROR DETECTED
                                CALL    QUIT            ;STOP THE XFR IN PROGRESS!
                                ERRB    TST1L,TST1CA
                                ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                TST1CA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                        ;>HOST CPU FAILED TO RESPOND TO 'SCLK' COMMAND WITH A 'WCLK' ON THE
                        ;>FIRST 'SCLK' ISSUED - DATA FAILED TO TRANSFER OVER MASSBUS
                        ;<TEST ABORTED!
                                JMP     TSTEND          ;ABORT NOW IF PROCEED ON ERROR SET
                        ;FINISH THE 'SCLK' CYCLE BY CLEARING THE 'SCLK' BIT IN DBUS CNTRL WORD

                        TST1CB: CALL    GETDDR          ;PULL DATA OFF MASSBUS VIA DDR REGS

                                MVI     A,M.OCC!T.SCLK  ;SAME STATUS MINUS THE 'SCLK' BIT
                                OUT     DBUSCTL         ;SET STATUS
                                ORI     M.RUN           ;ADD IN OTHER EXPECTED BITS
                                ROUT    EDATA           ;SAVE THE EXPECTED DATA
                                        OUT     EDATA           ;WRITE AC INTO EDATA
                                        MOV     A,A             ;RETRY LINK
                                NOP                     ;SMALL WAIT
                                IN      DBUSSTA         ;GET THE DBUS STATUS
                                ROUT    ADATA           ;SAVE ACTUAL DATA READ
```

```
  (1)  43AF  D3  94          10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  43B1  7F               4.0                    MOV     A,A             ;RETRY LINK
1589   43B2  E6  01           7.0            ANI     M.SCLK          ;SAVE ONLY ''SCLK'' STATUS BIT
1590   43B4  CA  C8  43       10.0            JZ      T1CHK           ;JUMP OVER IF THE BIT IS CLEARED
1591   43B7  CD  80  45       18.0            CALL    QUIT            ;ABORT XFR IF ERROR DETECTED!
1592   43BA                                   ERRB    TST1L,TST1CC
  (1)                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
  (1)  43BA  CD  12  28       18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        0004                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  43BD  04                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  43BE  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  43BF  CD  15  28       18.0                TST1CC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  43C2  DA  05  43       10.0                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1593                                       ;>EXPECTED CBUSSTA BYTE NOT = ACTUAL
1594                                       ;>''SCLK'' FAILED TO CLEAR AFTER IT WAS SET.
1595                                       ;<TEST ABORTED!
1596   43C5  C3  18  28           10.0            JMP     TSTEND
1597                                      ;DATA NOW TRANSFERED IN AN 18 BIT CHUNK FROM THE HOST CPU.
1598                                      ;CHECK THE DDR REGISTERS FOR EXPECTED RESULTS. REG ''B'' CONTAINS
1599                                      ;THE ''SCLK'' COUNT -1 (USED FOR A TABLE POINTER)
1600
1601   43C8  AF                    4.0  T1CHK:  XRA     A               ;CLEAR CAS REG 05 HIGH BYTE
1602   43C9                               ROUT    R05H            ;FOR ERROR PRINTING
  (1)  43C9  D3  8B           10.0                    OUT     R05H            ;WRITE AC INTO R05H
  (1)  43CB  7F                4.0                    MOV     A,A             ;RETRY LINK
1603   43CC  3A  53  46           13.0            LDA     CLKCNT          ;GET THE CLOCK COUNT
1604   43CF                               ROUT    R05L            ;SAVE IT IN REG 05 LOW BYTE FOR ERROR REPORT
  (1)  43CF  D3  8A           10.0                    OUT     R05L            ;WRITE AC INTO R05L
  (1)  43D1  7F                4.0                    MOV     A,A             ;RETRY LINK
1605   43D2  CD  D4  45           18.0            CALL    LD18BIT         ;GET THE EXPECTED 18 BIT DATA
1606   43D5  CD  AC  45           18.0            CALL    DDRACK          ;CHECK BITS 7 TO 0 (REG A)
1607   43D8  CA  EC  43           10.0            JZ      TST1D           ;JUMP IF NO ERROR DETECTED
1608   43DB  CD  80  45           18.0            CALL    QUIT            ;ABORT THE XFR NOW!
1609   43DE                                       ERRB    TST1L,TST1CD,6
  (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
  (1)  43DE  CD  12  28       18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        0005                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  43E1  05                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  43E2  06                                           .BYTE   6               ;PRINT ROUTINE NUMBER
  (1)  43E3  CD  15  28       18.0                TST1CD::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  43E6  DA  05  43       10.0                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1610                                       ;>MBD BITS 7-0 COMPARE ERROR ON THIS ''SCLK'' XFR
1611                                       ;<TEST ABORTED!
1612   43E9  C3  18  28           10.0            JMP     TSTEND
1613
1614                                       ;CHECK DATA IN BITS 15 TO 8 OF DDR (REG B)
1615
1616   43EC  CD  C3  45           18.0  TST1D:  CALL    DDRBCK          ;CHECK BITS 15 TO 8
1617   43EF  CA  03  44           10.0            JZ      TST1E           ;JUMP IF ALL OK
1618   43F2  CD  80  45           18.0            CALL    QUIT            ;ABORT THE XFR NOW
```

```
1620  43F5                                          ERRB    TST1L,TST1DC,6
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  43F5  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43F8  06                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43F9  06                                              .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  43FA  CD  15  28        18.0                  TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43FD  DA  05  43        10.0                          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1621                                            ;>MBD BITS 15-8 COMPARE ERROR ON THIS 'SCLK' XFR
1622  4400  C3  18  28        10.0                  JMP     TSTEND
1623                                            ;NOW TEST DATA IN DDR REG 'C' - PARITY BIT AND MASSBUS DATA BITS 17-16
1624                                            ;AND THE PARITY ERROR BITS.
1625
1626  4403  CD  FD  45        18.0  TST1E:  CALL    DDRCCK          ;CHECK DDR REG C
1627  4406  CA  1A  44        10.0          JZ      TST1F           ;JUMP IF NO ERROR DETECTED
1628  4409  CD  A1  45        18.0          CALL    QUIT1           ;ABORT WRITE XFR
1629  440C                                  ERRB    TST1L,TST1EC,6
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  440C  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  440F  07                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4410  06                                              .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  4411  CD  15  28        18.0                  TST1FC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4414  DA  05  43        10.0                          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1630                                            ;>MBD BITS 17-16 AND 'P' INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
1631                                            ;>OF WMC ERROR WORD - 100332).
1632  4417  C3  18  28        10.0          JMP     TSTEND          ;QUIT!
1633
1634  441A  DB  DA              10.0  TST1F:  IN      WMCERR          ;GET THE WMC ERROR BYTE
1635  441C  E6  40               7.0          ANI     M.PE            ;MASS BUSS PARITY ERROR?
1636  441E  CA  32  44          10.0          JZ      TST1G           ;NO - CONTINUE
1637  4421  CD  A1  45          18.0          CALL    QUIT1           ;TERMINATE THE MASS BUS TRANSFER
1638  4424                                  ERR     TST1L,TST1FC,6
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4424  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4427  08                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4428  06                                              .BYTE   6
 (1)  4429  CD  15  28        18.0                  TST1FC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  442C  DA  05  43        10.0                          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1639                                            ;>MASSBUS PARITY ERROR DETECTED BY THE TM78
1640                                            ;<TEST ABORTED!
1641  442F  C3  18  28        10.0          JMP     TSTEND          ;ABORT THE TEST SEQUENCE
1642                                            ;HERE WHEN TIME TO DO NEXT 'SCLK' CYCLE
1643  4432  3A  53  46          13.0  TST1G:  LDA     CLKCNT          ;GET CURRENT CLOCK COUNT
1644  4435  3C                   4.0          INR     A               ;UPDATE IT
1645  4436  32  53  46          13.0          STA     CLKCNT          ;SAVE THE UPDATED COUNTED
1646  4439  FE  29               7.0          CPI     41              ;DONE WITH THE TEST? (TOTAL OF 40 SCLKS)
1647  443B  CA  44  44          10.0          JZ      T1END           ;YES - TERMINATE TRANSFER AND EXIT
1648
1649  443E  CD  68  46          18.0          CALL    CLOCK0          ;NO - ISSUE NEXT 'SCLK' CYCLE
1650  4441  C3  C8  43          10.0          JMP     T1CHK           ;AND CHECK THE DATA IN DDR REGISTERS
```

```
1652                                        ;HERE WHEN THE TRANSFER IS COMPLETE - TERMINATE THE TRANSFER BY ISSUING
1653                                        ;MASSBUS 'EBL'
1654
1655  4444  3E  32          7.0  T1END:  MVI   A,M.OCC!M.EBL.T.SCLK  ;GET 'OCC'' + 'EBL'' + ''SCLK OUT''
1656  4446  D3  C0         10.0          OUT   DBUSCTL               ;SET THE STATUS - TERMINATE THE TRANSFER
1657  4448  3E  22          7.0          MVI   A,M.OCC!T.SCLK        ;DROP 'EBL'
1658  444A  D3  C0         10.0          OUT   DBUSCTL
1659  444C  3E  00          7.0          MVI   A,@0                  ;CLEAR DBUS CONTROL - ALL DONE WITH MASSBUS
1660  444E  D3  C0         10.0          OUT   DBUSCTL               ;RELEASE THE MASSBUS DATA LINES
1661  4450                               ENDTST  TST1L
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4450                              REQ   7                     ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4450  CD  06  28      18.0                        CALL  REQST
 (2)  4453  00                                          .BYTE                 ;DATA PATTERN NUMBER
 (2)  4454  00  00                                      .WORD                 ;SYSTEM '' COUNT
 (2)  4456  00  00                                      .WORD                 ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4458  00                                          .BYTE                 ;DATA COMPARE FLAG IF
 (2)  4459  07                                          .BYTE  7              ;REQUEST CODE
 (1)  445A  3A  9A  4F      13.0          LDA   ITERA                 ;GET ITERATION COUNT
 (1)  445D  3D               4.0          DCR   A                     ;DOWNCOUNT
 (1)  445E  32  9A  4F      13.0          STA   ITERA                 ;SAVE COUNT
 (1)  4461  F2  05  43      10.0          JP    TST1L                 ;DO TEST UNTIL TILL = 0
```

```
                        .SBTTL  TEST 2 - WRITE DATA TRANSFER USING THE ''DDR'' REGISTERS - WMC IN ''LEFT'' MODE
1664  4464              ST
 (1)                    ; ***************************************************************************
 (1)                    ;*TEST TITLE
 (1)                    ;*----- -----
1665                    ;*WRITE DATA TRANSFER USING THE ''DDR'' REGISTERS - WMC IN ''LEFT'' MODE
1666  4464              SM
 (1)                    ; ***************************************************************************
 (1)                    ;*MODULE(S) UNDER TEST
 (1)                    ;*--------------------
1667                    ;*M8956 & M8959
1668  4464              SD
 (1)                    ; ***************************************************************************
 (1)                    ;*DESCRIPTION
 (1)                    ;*-----------
1669                    ;*THIS TEST REQUIRES THE HOST CPU TO GENERATE THE ''MBCPAR'' DATA PATTERN
1670                    ;*TO SEND OVER THE MASSBUS ON A WRITE GCR COMMAND (FUNCTION CODE 63). IF
1671                    ;*THE HOST CPU IS NOT CAPABLE OF AN 18 BIT DATA TRANSFER OVER THE
1672                    ;*MASSBUS, THEN THE TEST EXPECTS A FUNCTION CODE OF 65 TO BE RETURNED!
1673                    ;*
1674                    ;*THE HOST WILL BE REQUESTED TO TRANSFER 100 BYTES (18 BIT) OF DATA USING THE
1675                    ;*10 DUMP MODE PACKING FORMAT (5 BYTES PER 10 CPU WORD). THE BYTE COUNT
1676                    ;*REQUESTED WILL APPEAR IN THE MASSBUS REGISTER 05 (BYTE COUNT REGISTER).
1677                    ;*
1678                    ;*MASSBUS REGISTER 02 WILL CONTAIN THE RECORD COUNT (SET TO 1) AND THE
1679                    ;*PACKING MODE TO USE (CODE =2 FOR 10 DUMP) - NO UNIT IS NEEDED FOR THE
1680                    ;*TEST.
1681                    ;*
1682                    ;*THIS TEST WILL ISSUE 40 ''SCLK'' CYCLES TO TRANSFER THE DATA FROM THE
1683                    ;*HOST CPU. DATA IS CHECKED AFTER EACH CLOCK CYCLE AND WILL CAUSE AN
1684                    ;*ERROR REQUEST TO THE HOST IF FOUND TO BE INCORRECT!
1685                    ;*
1686                    ;*DETECTION OF A MASSBUS DATA PARITY ERROR OR ''OTHER'' ERROR STATUS IN
1687                    ;*THE WMC ERROR STATUS WORD (100332) ALSO CAUSES AN ERROR REPORT.
1688                    ;*
1689                    ;*DATA ERRORS OR STATUS ERRORS WILL TERMINATE THE DATA TRANSFER IF HOST
1690                    ;*CPU HAS REQUESTED THE MICROCOMPUTER TO ''LOOP-ON-ERROR''. INTERNAL STATUS
1691                    ;*ERRORS WILL BE CLEARED IF REPORTED TO BE SET.
1692                    ;*
1693                    ;*MASSBUS DATA TEST - USE HOST CPU TO CHECK MBD MODULE OK FOR WRITE
1694                    ;*AND READ DATA TRANSFERS.
1695                    ;*WRITE XFR TEST
1696  4464              SP
 (1)                    ; ***************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
1697                    ;*BGNTST
1698                    ;*   INIT TEST - TESTX MACRO
1699                    ;*   SET WMC 'LEFT' MODE TO TEST EXTRA BITS
1700                    ;*   ISSUE WMC RESTART TO INIT WRITE PATH
1701                    ;*   IF WMC 'LEFT' SET
1702                    ;*   :  THEN-CONTINUE
1703                    ;*   :  ELSE-REPORT ERROR
1704                    ;*   ENDIF
```

```
                                ;*  ISSUE WMC RESTART TO INIT WRITE PATH
                                ;*  INIT ECODE COUNTER TO 0 AND SELECT WMC CONTROL TO 'WRITE'
                                ;*  SET DATA BUS CONTROL TO 0 + WAIT SMALL AMOUNT OF TIME
                                ;*  IF DBUS STATUS STILL=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT ERROR - INCORRECT DBUS
                                ;*  :       STATUS - MAYBE CABLE FAILURE
                                ;*  ENDIF
                                ;*  INIT DDR REGS TO ''IN'' + REQUEST HOST CPU FOR A 'WRITE' GCR DATA XFR
                                ;*  CONTINUE WITH HOST RESPONSE - CODE 63 OR 67 (IF HOST NOT CAPABLE OF 18 BIT XFR)
                                ;*  DATA SENT BY HOST IS 'MBDPAR' IN 10 DUMP MODE FORMAT
                                ;*  INIT 'SCLK' COUNT MASSBUS CONTROL TO 'OCC'-'RUN'-'MBRDEN'
                                ;*  SET 'SCLK' + WAIT FOR HOST RESPONSE WITH 'WCLK'
                                ;*  IF 'WCLK' RESPONSE BY HOST
                                ;*  :  THEN-REPORT NO HOST WCLK' RESPONSE-OPI ERROR
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  FINISH 1ST 18 BIT 'SCLK' CYCLE BY CLEARING 'SCLK' BIT FROM DBUS CONTROL
                                ;*  IF 'SCLK' GO AWAY
                                ;*  :  THEN-REPORT INCORRECT DBUS STATUS
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  GET DATA IN DDR REGS 'A', 'B', + 'C' + CHECK IF DATA OK
                                ;*  IF DATA RECEIVED OK
                                ;*  :  THEN-REPORT READ FAILURE OF DATA FROM HOST CPU
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  DO UNTIL ALL DATA CLOCKED FROM HOST AND CHECKED
                                ;*  :  CLOCK NEXT 18 BITS ACROSS MASSBUS
                                ;*  :  IF DATA RECEIVED OK
                                ;*  :  :  THEN-CONTINUE
                                ;*  :  :  ELSE-REPORT DATA XFR ERROR + 'SCLK' COUNT
                                ;*  :  ENDIF
                                ;*  ENDDO
                                ;*  DROP MASSBUS SIGNALS 'OCC' + 'RUN'' TO RELEASE THE HOST CPU
                                ;*ENDTST
 4464                           SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MBD1 MICRO TEST 02
                                ;*MBD1 MICRO ERROR 11
                                ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-'DDR'' IN 'LEFT'' MODE
                                ;*M8956, M8959
                                ;*WMC FAILED TO SET 'WMC LEFT'' STATUS BIT FOR DDR CONTROL NOT SET
                                ;*ACTUAL = NNNN
                                ;*TEST ABORTED!
                                ;*
                                ;*MBD1 MICRO TEST 02
                                ;*MBD1 MICRO ERROR 12
                                ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-'DDR'' IN 'LEFT'' MODE
                                ;*M8956, M8959
                                ;*OPI ERROR - HOST CPU FAILED TO SET 'RUN'' ON A WRITE COMMAND (6X CODE)
```

```
                                        ;*TEST ABORTED!
                                        ;*
                                        ;*MBD1 MICRO TEST 02
                                        ;*MBD1 MICRO ERROR 13
                                        ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-'DDR' IN 'LEFT' MODE
                                        ;*M8956, M8959
                                        ;*MBD BITS 7-0 COMPARE ERROR ON THIS 'SCLK' XFR
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*EXPECTED 18 BITS = E EEEEEE
                                        ;*ACTUAL 18 BITS   = A AAAAAA
                                        ;*
                                        ;*MBD1 MICRO TEST 02
                                        ;*MBD1 MICRO ERROR 14
                                        ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-'DDR' IN 'LEFT' MODE
                                        ;*M8956, M8959
                                        ;*MBD BITS 15-8 COMPARE ERROR ON THIS 'SCLK' XFR
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*EXPECTED 18 BITS = E EEEEEE
                                        ;*ACTUAL 18 BITS   = A AAAAAA
                                        ;*TEST ABORTED!
                                        ;*
                                        ;*MBD1 MICRO TEST 02
                                        ;*MBD1 MICRO ERROR 15
                                        ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-'DDR' IN 'LEFT' MODE
                                        ;*M8956, M8959
                                        ;*MBD BITS 17-16 AND 'P' INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
                                        ;*OF WMC ERROR WORD - 100332).
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED - NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*EXPECTED 18 BITS = E EEEEEE
                                        ;*ACTUAL 18 BITS   = A AAAAAA
                                        ;*
                                        ;*MBD1 MICRO TEST 02
                                        ;*MBD1 MICRO ERROR 16
                                        ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-'DDR' IN 'LEFT' MODE
                                        ;*M8956, M8959
                                        ;*MASSBUS PARITY ERROR DETECTED BY THE TM78
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*EXPECTED 18 BITS = E EEEEEE
                                        ;*ACTUAL 18 BITS   = A AAAAAA
                                        ;*TEST ABORTED!
1802  4464                              S
 (1)                                    ; ************************************************************************************
1803  4464                              TEST2:  TESTX   @2
 (1)  4464  3E  02              7.0                 MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4466  CD  03  28         18.0                 CALL    TSET            ;SETUP THE TEST
1804                                    ;%MBD1-DATA WRITE TM78 TEST USING DIAG. DATA REG.-'DDR' IN 'LEFT' MODE
1805                                    ;&M8956, M8959
1806
```

```
1807                                        ;NOW SET THE WMC ''LEFT'' BIT
1808
1809  4469  3E  88            7.0  TST2L:  MVI   A,@210          ;PUT DDR IN THE ''IN'' MODE
1810  446B  D3  DB           10.0          OUT   DDRCTL
1811  446D  3E  39            7.0          MVI   A,@71           ;PUT A FMT TYPE OF ''10 DUMP'' & SKP CNT-11
1812  446F  D3  D0           10.0          OUT   DATACTL         ;IN CONTROL REG FOR WMC PGM
1813  4471  3E  01            7.0          MVI   A,W.RST         ;RESTART THE WMC
1814  4473  D3  D3           10.0          OUT   WMCCTL
1815
1816                                        ;CLEAR THE ERROR CODE COUNTER
1817
1818  4475                                  CLRECT
 (1)  4475  AF                 4.0                      XRA   A               ;CLEAR THE ACCUMULATOR
 (1)  4476  D3  D6            10.0                      OUT   ERRCNT          ;CLEAR BITS 7-0
 (1)  4478  D3  D6            10.0                      OUT   ERRCNT          ;CLEAR BITS 15-8
1819  447A  AF                 4.0          XRA   A               ;FINISH THE RESTART CYCLE
1820  447B  D3  D3            10.0          OUT   WMCCTL
1821  447D  3E  80             7.0          MVI   A,W.ENAB        ;THEN TELL WMC TO SET ''LEFT''
1822  447F  D3  D3            10.0          OUT   WMCCTL
1823
1824  4481  3E  28             7.0          MVI   A,40
1825  4483  3D                 4.0  T2WS:   DCR   A               ;WAIT FOR ABOUT 100 MICRO-SEC TO SET
1826  4484  C2  83  44        10.0          JNZ   T2WS            ;DELAY TILL TIMEOUT
1827
1828                                        ;NOW CHECK THAT THE WMC PGM REALLY SET ''WMC LEFT'' BIT SO LOADING DDRA-DDRC
1829                                        ;WILL BE DIRECT TRANSLATION TO MASSBUS BITS.
1830
1831  4487  DB  A1           10.0          IN    CBUSSTA         ;GET CBUS STATUS (WMC LEFT BIT)
1832  4489                                  ROUT  ADATA           ;SAVE ACTUAL STATUS FOR ERROR REPORT
 (1)  4489  D3  94            10.0                      OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  448B  7F                 4.0                      MOV   A,A             ;RETRY LINK
1833  448C  E6  04             7.0          ANI   W.LEFT          ;SAVE ONLY WMC LEFT BIT
1834  448E  C2  9F  44        10.0          JNZ   TST2GO          ;CONTINUE IF ITS SET...
1835  4491                                  ERRA  TST2L,TST2LC
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4491  CD  0F  28        18.0                      CALL  ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0009                                        MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4494  09                                          .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4495  00                                          .BYTE                 ;PRINT ROUTINE NUMBER
 (1)  4496  CD  15  28        18.0          TST2LC::                CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4499  DA  69  44        10.0                      JC    TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1836                                        ;>WMC FAILED TO SET ''WMC LEFT'' STATUS BIT FOR DDR CONTROL NOT SET
1837                                        ;<TEST ABORTED!
1838  449C  C3  18  28        10.0          JMP   TSTEND          ;ABORT RIGHT NOW!
1839  449F  3E  08             7.0  TST2GO: MVI   A,W.WRITE      ;ENABLE ONLY ''WRITE'' IN WMC CNTRL WORD
1840  44A1  D3  D3            10.0          OUT   WMCCTL          ;SET WRITE STATUS
1841
1842                                        ;PROCEED TO SETUP THE DDR REGISTERS (CLEAR THEM)
1843
1844  44A3  3E  9B             7.0  TST2B:  MVI   A,@233          ;SET DDR CNTRL WORD TO ''IN''
1845  44A5  D3  DB            10.0          OUT   DDRCTL
1846                                        ;MASSBUS REGISTERS 02 (RECORD COUNT, FMT TYPE, SKIP CNT) AND REGISTER
1847                                        ;05 (BYTE COUNT REG) WILL BE SET UP ON THE DIAGNOSTIC MONITOR CALL.
1848
```

```
                                        ;REQUEST THE DIAG. MON. TO SIGNAL THE HOST CPU FOR A WRITE GCR DATA
                                        ;TRANSFER USING 'MBDPAR'' DATA PATTERN (2ND ARG IN CALL). BYTE COUNT
                                        ;IS 100. (40 SCLK * 2.5 BYTES PER WORD IN 10 DUMP MODE). REG 02 GETS
                                        ;LOADED TO 1 RECORD @ 10 DUMP FORMAT.

44A7                                            REQ     2,1,40,@30004
 (1) 44A7  CD  06  28        18.0                       CALL    REQST
 (1) 44AA  01                                           .BYTE   1               ;DATA PATTERN NUMBER
 (1) 44AB  00  28                                       .WORD   40              ;SYSTEM ''40'' COUNT
 (1) 44AD  30  04                                       .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 44AF  00                                           .BYTE                   ;DATA COMPARE FLAG IF -1
 (1) 44B0  02                                           .BYTE   2               ;REQUEST CODE

                                        ;INIT THE SCLK COUNTER TO 0 ... TEST WILL STOP WHEN COUNT = 41.

44B1  3E  01                 7.0                MVI     A,1
44B3  32  53  46            13.0                STA     CLKCNT          ;INIT THE CLOCK COUNTER

                                        ;SET THE DBUS CNTRL WORD TO ''OCC'', AND ''SCLK OUT''

44B6  3E  22                 7.0                MVI     A,M.OCC.T.SCLK
44B8  D3  C0                10.0                OUT     DBUSCTL         ;SET THE STATUS - NOW HAVE MASSBUS

                                        ;NOW WAIT FOR HOST CPU TO ASSERT ''RUN'' ON THE MASSBUS...COULD TAKE UP TO
                                        ;10 MILLISEC (MASSBUS SPEC.)

44BA  0E  14                 7.0                MVI     C,20            ;LOOK FOR 100 M-SEC.
44BC  DB  C0                10.0  T2W:    IN      DBUSSTA         ;GET DBUS STATUS - CHECK FOR ''RUN''
44BE  E6  04                 7.0                ANI     M.RUN           ;SAVE ONLY ''RUN'' BIT
44C0  C2  DE  44            10.0                JNZ     TST2C           ;JUMP IF RUN IS UP - OK TO PROCEED
44C3  3E  FF                 7.0  T2WL:   MVI     A,@377          ;GET A DELAY TIME
44C5  3D                     4.0  T2WL1:  DCR     A               ;DECREMENT
44C6  C2  C5  44            10.0                JNZ     T2WL1           ;STAY IN LOOP TILL =0
                                        ;DOWNCOUNT GROSS TIMER COUNT

44C9  0D                     4.0                DCR     C
44CA  C2  BC  44            10.0                JNZ     T2W             ;THEN CHECK IF ''RUN'' IS UP
44CD  CD  80  45            18.0                CALL    QUIT
44D0                                            ERR     TSTEND,T2WLC
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 44D0  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000A                                         MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 44D3  0A                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 44D4  00                                           .BYTE
 (1) 44D5  CD  15  28        18.0               T2WLC:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 44D8  DA  18  28        10.0                       JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>OPI ERROR - HOST CPU FAILED TO SET ''RUN'' ON A WRITE COMMAND (6X CODE)
                                        ;<TEST ABORTED!
44DB  C3  18  28            10.0                JMP     TSTEND          ;FATAL ERROR. - QUIT NOW

                                        ;START THE CLOCK CYCLE (SCLK) BY SETTING THE SCLK BIT IN DBUS CNTRL
                                        ;WORD

44DE  CD  68  46            18.0  TST2C:  CALL    CLOCK0          ;CLOCK THE 1ST 18 BITS OF DATA TO TM78
```

```
                                        ;DATA NOW TRANSFERED IN AN 18 BIT CHUNK FROM THE HOST CPU.
                                        ;CHECK THE DDR REGISTERS FOR EXPECTED RESULTS. REG "B" CONTAINS
                                        ;THE "SCLK" COUNT -1 (USED FOR A TABLE POINTER)

44E1  AF                        4.0     T2CHK:  XRA     A               ;CLEAR CAS REG 05 HIGH BYTE
44E2                                            ROUT    R05H            ;FOR ERROR PRINTING
 (1)  44E2  D3  8B              10.0                    OUT     R05H            ;WRITE AC INTO R05H
 (1)  44E4  7F                   4.0                    MOV     A,A             ;RETRY LINK
44E5  3A  53  46                13.0            LDA     CLKCNT          ;GET THE CLOCK COUNT
44E8                                            ROUT    R05L            ;SAVE IT IN REG 05 LOW BYTE FOR ERROR REPORT
 (1)  44E8  D3  8A              10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  44EA  7F                   4.0                    MOV     A,A             ;RETRY LINK
44EB  CD  D4  45                18.0            CALL    LD18BIT         ;GET THE EXPECTED DATA
44EE  CD  AC  45                18.0            CALL    DDRACK          ;CHECK BITS 7 TO 0 (REG A)
4501  CA  05  45                10.0            JZ      TST2D           ;JUMP IF NO ERROR DETECTED
4504  CD  80  45                18.0            CALL    QUIT            ;ABORT THE XFR NOW!
4507                                            ERRB    TST2L,TST2CD,6
                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)
 (1)  44F7  CD  12  28          18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44FA  08                                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44FB  06                                                  .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  44FC  CD  15  28          18.0                    TST2CD::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FF  DA  69  44          10.0                            JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>MBD BITS 7-0 COMPARE ERROR ON THIS "SCLK" XFR
4502  C3  18  28                10.0            JMP     TSTEND

                                        ;CHECK DATA IN BITS 15 TO 8 OF DDR (REG B)

4505  CD  C3  45                18.0    TST2D:  CALL    DDRBCK          ;CHECK BITS 15 TO 8
4508  CA  1C  45                10.0            JZ      TST2E           ;JUMP IF ALL OK
450B  CD  80  45                18.0            CALL    QUIT            ;ABORT THE XFR NOW
450E                                            ERRB    TST2L,TST2DC,6
                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)
 (1)  450E  CD  12  28          18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000C                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4511  0C                                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4512  06                                                  .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  4513  CD  15  28          18.0                    TST2DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4516  DA  69  44          10.0                            JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>MBD BITS 15-8 COMPARE ERROR ON THIS "SCLK" XFR
                                        ;<TEST ABORTED!
4519  C3  18  28                10.0            JMP     TSTEND
                                        ;NOW TEST DATA IN DDR REG "C" - PARITY BIT AND MASSBUS DATA BITS 17-16
                                        ;AND THE PARITY ERROR BITS.

451C  CD  FD  45                18.0    TST2E:  CALL    DDRCCK          ;CHECK DDR REG C
451F  CA  33  45                10.0            JZ      TST2F           ;JUMP IF NO ERROR DETECTED
4522  CD  A1  45                18.0            CALL    QUIT1           ;ABORT WRITE XFR
```

```
1922  4525                                          ERRB    TST2L,TST2EC,6
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4525  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000D                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4528  0D                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4529  06                                            .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  452A  CD  15  28        18.0                TST2EC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  452D  DA  69  44        10.0                        JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1923                                            ;>MBD BITS 17-16 AND "P" INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
1924                                            ;>OF WMC ERROR WORD - 100332).
1925  4530  C3  18  28        10.0                JMP     TSTEND          ;QUIT!
1926
1927  4533  DB  DA              10.0  TST2F:  IN      WMCERR          ;GET THE WMC ERROR BYTE
1928  4535  E6  40               7.0          ANI     M.PE            ;MASS BUSS PARITY ERROR?
1929  4537  CA  4B  45          10.0          JZ      TST2G           ;NO - CONTINUE
1930  453A  CD  A1  45          18.0          CALL    QUIT1           ;TERMINATE THE MASS BUS TRANSFER
1931  453D                                    ERR     TST2L,TST2FC,6
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453D  CD  09  28        18.0                          CALL    ERLP            PROCESS ERROR - DO 2.3
 (1)       000E                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4540  0E                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4541  06                                            .BYTE   6
 (1)  4542  CD  15  28        18.0                TST2FC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4545  DA  69  44        10.0                        JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1932                                            ;>MASSBUS PARITY ERROR DETECTED BY THE TM78
1933                                            ;<TEST ABORTED!
1934  4548  C3  18  28        10.0                JMP     TSTEND          ;ABORT THE TEST SEQUENCE
1935                                            ;HERE WHEN TIME TO DO NEXT "SCLK" CYCLE
1936
1937  454B  3A  53  46          13.0  TST2G:  LDA     CLKCNT          ;GET CURRENT CLOCK COUNT
1938  454E  3C                   4.0          INR     A               ;UPDATE IT
1939  454F  32  53  46          13.0          STA     CLKCNT          ;SAVE THE UPDATED COUNTED
1940  4552  FE  29               7.0          CPI     41              ;DONE WITH THE TEST?
1941  4554  CA  5D  45          10.0          JZ      T2END           ;YES - TERMINATE TRANSFER AND EXIT
1942
1943  4557  CD  68  46          18.0          CALL    CLOCK0          ;NO - ISSUE NEXT "SCLK" CYCLE
1944  455A  C3  E'  44          10.0          JMP     T2CHK           ;AND CHECK THE DATA IN DDR REGISTERS
1945
1946                                            ;HERE WHEN THE TRANSFER IS COMPLETE - TERMINATE THE TRANSFER BY ISSUING
1947                                            ;MASSBUS "EBL"
1948
1949  455D  3E  32               7.0  T2END:  MVI     A,M.OCC!M.EBL.T.SCLK  ;GET "OCC" + "EBL" + "SCLK OUT"
1950  455F  D3  C0              10.0          OUT     DBUSCTL         ;SET THE STATUS - TERMINATE THE TRANSFER
1951  4561  3E  22               7.0          MVI     A,M.OCC!T.SCLK  ;DROP "EBL"
1952  4563  D3  C0              10.0          OUT     DBUSCTL
1953  4565  3E  00               7.0          MVI     A,@0            ;CLEAR DBUS CONTROL - ALL DONE WITH MASSBUS
1954  4567  D3  C0              10.0          OUT     DBUSCTL         ;RELEASE THE MASSBUS DATA LINES
```

```
                                        ENDTST  TST2L
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
(2) 4569
(2) 4569 CD 06 28       18.0                            CALL    REQST
(2) 456C 00                                             .BYTE                   ;DATA PATTERN NUMBER
(2) 456D 00 00                                          .WORD                   ;SYSTEM '*' COUNT
(2) 456F 00 00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2) 4571 00                                             .BYTE                   ;DATA COMPARE FLAG IF =1
(2) 4572 07                                             .BYTE   7               ;REQUEST CODE
(1) 4573 3A 9A 4F       13.0            LDA     ITERA           ;GET ITERATION COUNT
(1) 4576 3D              4.0            DCR     A               ;DOWNCOUNT
(1) 4577 32 9A 4F       13.0            STA     ITERA           ;SAVE COUNT
(1) 457A F2 69 44       10.0            JP      TST2L           ;DO TEST UNTIL TILL = 0
1957 457D C3 FB 47      10.0            JMP     TEST3           ;DO THE NEXT TEST
1958                            ;HERE IF A DATA COMPARE ERROR DETECTED ON THE DDR REGISTERS AND
1959                            ;HOST CPU REQUESTED TO 'LOOP-ON-ERROR'
1960
1961 4580 3E 0A          7.0    QUIT:   MVI     A,M.EXC!T.SCLK  ;SEND MASSBUS 'EXC'
1962 4582 D3 C0         10.0            OUT     DBUSCTL         ;TO TERMINATE HOST XFR
1963 4584 3E 1A          7.0            MVI     A,M.EBL!M.EXC!T.SCLK  ;SET BOTH EXC AND EBL
1964 4586 D3 C0         10.0            OUT     DBUSCTL
1965 4588 3E 00          7.0            MVI     A,0
1966 458A D3 C0         10.0            OUT     DBUSCTL         ;RELEASE THE MASSBUS
1967 458C 3A 54 46      13.0            LDA     SDDRA           ;GET ACTUAL 7-0
1968 458F                               ROUT    R16L
(1) 458F D3 9C          10.0                    OUT     R16L            ;WRITE AC INTO R16L
(1) 4591 7F              4.0                    MOV     A,A             ;RETRY LINK
1969 4592 3A 55 46      13.0            LDA     SDDRB           ;GET ACTUAL 15-8
1970 4595                               ROUT    R16H
(1) 4595 D3 9D          10.0                    OUT     R16H            ;WRITE AC INTO R16H
(1) 4597 7F              4.0                    MOV     A,A             ;RETRY LINK
1971 4598 3A 56 46      13.0            LDA     SDDRC           ;GET ACTUAL 17-16
1972 459B F6 80          7.0            ORI     BIT7
1973 459D                               ROUT    R17L
(1) 459D D3 9E          10.0                    OUT     R17L            ;WRITE AC INTO R17L
(1) 459F 7F              4.0                    MOV     A,A             ;RETRY LINK
1974 45A0 C9            10.0            RET                     ;EXIT
1975                            ;HERE IS CODE NECESSARY TO CLEAR PARITY ERROR OR 'OTHER' ERRORS DETECTED
1976                            ;IN WMC ERROR STATUS (ALSO DDR REG 'C').
1977
1978 45A1 3E 09          7.0    QUIT1:  MVI     A,W.WRITE!W.RST ;SET 'WRITE' + 'RESTART'
1979 45A3 D3 D3         10.0            OUT     WMCCTL          ;IN WMC CONTROL WORD
1980 45A5 3E 08          7.0            MVI     A,W.WRITE       ;RESET TO ORIG. STATUS OF 'WRITE'
1981 45A7 D3 D3         10.0            OUT     WMCCTL
1982 45A9 C3 80 45      10.0            JMP     QUIT            ;NORMAL ERROR TERMINATION OF DATA TRANSFER
1983 45AC                       S
(1)                             ; *********************************************************************************
1984                            ;DDRACK -- ROUTINE TO CHECK THE CONTENTS OF DDR REG A (DATA BITS 7-0)
1985 45AC                       S
(1)                             ; *********************************************************************************
1986
1987 45AC C5            12.0    DDRACK: PUSH    B               ;SAVE B & C
1988 45AD 32 51 46      13.0            STA     ASAV            ;SAVE DATA IN REG A
1989 45B0 3A 5A 46      13.0            LDA     SDDRAE          ;GET EXPECTED DATA
```

```
                                        ROUT    EDATA
                                        OUT     EDATA           ;WRITE AC INTO EDATA
                                        MOV     A,A             ;RETRY LINK
                                MOV     B,A             ;TEMP SAVE IT
                                LDA     SDDRA           ;GET THE READ DDRA DATA
                        COMCK:  ROUT    ADATA           ;SAVE IN CASE OF ERROR
                                        OUT     ADATA           ;WRITE AC INTO ADATA
                                        MOV     A,A             ;RETRY LINK
                                CMP     B               ;IF ERROR DETECTED, "Z" WILL =0
                                POP     B
                                LDA     ASAV            ;RESET REG A
                                RET

                        S
                        ; ***************************************************************************
                        ;DDRBCK -- ROUTINE TO CHECK THE CONTENTS OF DDR REG B (DATA BITS 15-8)
                        S
                        ; ***************************************************************************

                        DDRBCK: PUSH    B               ;SAVE B & C
                                STA     ASAV            ;SAVE REG A
                                LDA     SDDRBE          ;GET EXPECTED DATA
                                ROUT    EDATA           ;SAVE EXPECTED
                                        OUT     EDATA           ;WRITE AC INTO EDATA
                                        MOV     A,A             ;RETRY LINK
                                MOV     B,A             ;TEMP SAVE
                                LDA     SDDRB           ;GET THE "ADATA" FROM DDR REG B
                                JMP     COMCK           ;COMMON EXIT
                        S
                        ; ***************************************************************************
                        ;LD18BIT -- ROUTINE TO LOAD THE 'EXPECTED' DDR DATA STORAGE AREA FROM
                        ;       THE CLOCK COUNT IN REG A.
                        S
                        ; ***************************************************************************

                        LD18BIT: PUSH   B               ;SAVE B & C
                                PUSH    H               ;AND H & L
                                DCR     A
                                MOV     B,A             ;COPY THE COUNT
                                ADD     A               ;CALC. THE DATA TABLE OFFSET
                                ADD     B               ;REALLY = COUNT * 3
                                MOV     C,A             ;PUT RESULT IN REG C
                                MVI     B,0             ;B & C POINT TO DATA TABLE ENTRY
                                LXI     H,DATATB        ;GET START OF TABLE ADDRESS
                                DAD     B               ;H & L NOW POINT TO ENTRY ADDRESS
                                MOV     A,M             ;GET DDRA EXPECTED
                                STA     SDDRAE          ;SAVE IT
                                ROUT    R14L            ;SAVE BITS 7-0 FOR PRINTING EXPECTED
                                        OUT     R14L            ;WRITE AC INTO R14L
                                        MOV     A,A             ;RETRY LINK
                                INX     H               ;POINT TO NEXT
                                MOV     A,M             ;GET EXPECTED DDRB
                                STA     SDDRBE          ;SAVE IT
                                ROUT    R14H            ;SAVE BITS 15-8 FOR PRINTING EXPECTED
```

```
 (1)  45ED  D3  99          10.0                    OUT     R14H            ;WRITE AC INTO R14H
 (1)  45EF  7F               4.0                    MOV     A,A             ;RETRY LINK
2032  45F0  23               6.0            INX     H                       ;POINT TO NEXT
2033  45F1  7E               7.0            MOV     A,M                     ;GET EXPECTED DDRC
2034  45F2  32  5C  46       13.0           STA     SDDRCE                  ;SAVE IT
2035  45F5  F6  80            7.0           ORI     BIT7                    ;SET THE SIGN BIT
2036  45F7                                  ROUT    R15L                    ;SAVE BITS 17-16 FOR PRINTING EXPECTED
 (1)  45F7  D3  9A          10.0                    OUT     R15L            ;WRITE AC INTO R15L
 (1)  45F9  7F               4.0                    MOV     A,A             ;RETRY LINK
2037  45FA  E1               10.0           POP     H                       ;RESET REGS
2038  45FB  C1               10.0           POP     B
2039  45FC  C9               10.0           RET                             ;EXIT
2040  45FD                                  S
 (1)                                          ;*****************************************************************************
2041                                          ;DDRCCK -- ROUTINE TO CHECK THE CONTENTS OF DDR REG C (DATA BITS 17-16)
2042                                          ;       AND THE PARITY BIT ''P'' WITH THE MASSBUS DATA PARITY ERROR FLAG
2043                                          ;       AND ''OTHER'' PARITY ERROR STATUS.
2044  45FD                                  S
 (1)                                          ;*****************************************************************************
2045
2046  45FD  C5               12.0   DDRCCK: PUSH    B                       ;SAVE REG B & C
2047  45FE  32  51  46       13.0           STA     ASAV                    ;SAVE REG A
2048  4601  3A  5C  46       13.0           LDA     SDDRCE                  ;GET EXPECTED DDRC DATA
2049  4604                                  ROUT    EDATA                   ;SAVE IN CASE ERROR DETECTED
 (1)  4604  D3  95          10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4606  7F               4.0                    MOV     A,A             ;RETRY LINK
2050  4607  47               4.0            MOV     B,A                     ;TEMP SAVE
2051  4608  3A  56  46       13.0           LDA     SDDRC                   ;GET REG C DATA
2052  460B  E6  40            7.0           ANI     @100                    ;SEE IF ANY DATA PARITY ERROR STATUS DETECTED
2053  460D  C2  5D  46       10.0           JNZ     DDRCER                  ;ERROR DETECTED!!!
2054  4610  3A  56  46       13.0           LDA     SDDRC                   ;GET DDR REG C AGAIN
2055  4613  E6  07            7.0           ANI     7                       ;SAVE ONLY ''P'' AND 17-16
2056  4615  32  56  46       13.0           STA     SDDRC                   ;SAVE ONLY DATA IN DDRC AREA
2057  4618  B8               4.0            CMP     B                       ;SAME AS EXPECTED?
2058  4619  C2  21  46       10.0           JNZ     DDRCED                  ;NO - POSSIBLE HOST NOT CAPABLE...
2059
2060                                        ;HERE TO EXIT DDR REG C CHECK
2061
2062  461C  C1               10.0   DDRCEX: POP     B                       ;RESET REGS
2063  461D  3A  51  46       13.0           LDA     ASAV
2064  4620  C9               10.0           RET                             ;EXIT
2065                                        ;HERE IF REG C DATA ERROR DETECTED OR HOST NOT CAPABLE ERROR
2066
2067  4621  3A  98  4F       13.0   DDRCED: LDA     VALFC                   ;GET THE FUNCTION CODE RETURNED FROM REQ
2068  4624  FE  35            7.0           CPI     @65                     ;SEE IF HOST ''NOT CAPABLE'' IS SET
2069  4626  C2  1C  46       10.0           JNZ     DDRCEX                  ;NOT SET - MUST BE REAL ERROR.
2070
2071  4629  78               4.0            MOV     A,B                     ;GET EXPECTED DATA FOR DDRC AGAIN
2072  462A  E6  04            7.0           ANI     4                       ;SAVE ONLY ''P''
2073  462C  32  52  46       13.0           STA     PBIT                    ;SAVE THE CONDITION OF ''P'' DATA
2074  462F  78               4.0            MOV     A,B                     ;GET DATA AGAIN
2075  4630  E6  03            7.0           ANI     3                       ;SAVE ONLY 17 & 16 DATA BITS
2076  4632  CA  43  46       10.0           JZ      PNMOD                   ;NO MODIFICATION TO ''P'' BIT
2077  4635  FE  03            7.0           CPI     3                       ;IS IT -3 (NO MOD IF YES)
```

```
2078  4637  CA  43  46      10.0          JZ      PNMOD
2079  463A  3A  52  46      13.0          LDA     PBIT            ;GET THE CURRENT PBIT CONDITION
2080  463D  2F                4.0          CMA                     ;COMPLIMENT IT
2081  463E  E6  04            7.0          ANI     @4              ;SAVE ONLY THE P BIT AGAIN
2082  4640  32  52  46      13.0          STA     PBIT            ;SAVE
2083
2084  4643  3A  52  46      13.0  PNMOD:  LDA     PBIT            ;GET IT AGAIN
2085  4646                           ROUT    EDATA           ;UPDATE "EDATA" STORED
 (1)  4646  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4648  7F                4.0                   MOV     A,A             ;RETRY LINK
2086  4649  47                4.0          MOV     B,A             ;SAVE NEW VALUE OF EXPECTED DDRC DATA
2087  464A  3A  56  46      13.0          LDA     SDDRC           ;GET DDP REG C AGAIN
2088  464D  B8                4.0          CMP     B               ;OK NOW?
2089  464E  C3  1C  46      10.0          JMP     DDRCEX          ;EXIT - "Z" SET IF OK
2090
2091  4651  00                       ASAV:   .BYTE   0               ;STORAGE FOR REG A
2092  4652  00                       PBIT:   .BYTE   0               ;STORAGE FOR MODIFIED DDR REG "C"
2093  4653  00                       CLKCNT: .BYTE   0               ;COPY OF THE SCLK COUNT IN PROGRESS
2094  4654  00                       SDDRA:  .BYTE   0               ;STORAGE FOR DDRA DATA
2095  4655  00                       SDDRB:  .BYTE   0               ;STORAGE FOR DDRB DATA
2096  4656  00                       SDDRC:  .BYTE   0               ;STORAGE FOR DDRC DATA
2097  4657  00                       SDDRAT: .BYTE   0               ;TEMP STORE DDRA (REAL DDRA)
2098  4658  00                       SDDRBT: .BYTE   0               ;TEMP STORE DDRB (REAL DDRB)
2099  4659  00                       SDDRCT: .BYTE   0               ;TEMP STORE DDRC (REAL DDRC)
2100  465A  00                       SDDRAE: .BYTE   0               ;EXPECTED DDRA DATA STORAGE
2101  465B  00                       SDDRBE: .BYTE   0               ;EXPECTED DDRB
2102  465C  00                       SDDRCE: .BYTE   0               ;EXPECTED DDRC
2103                                 ;HERE IF PARITY ERROR OR OTHER STATUS ERROR DETECTED IN WMC ERROR WORD
2104
2105  465D  3A  56  46      13.0  DDRCER: LDA     SDDRC           ;GET DDR REG C AGAIN
2106  4660                           ROUT    ADATA           ;SAVE AS "ADATA" FOR ERROR REPORT
 (1)  4660  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4662  7F                4.0                   MOV     A,A             ;RETRY LINK
2107  4663  E6  FF            7.0          ANI     @377
2108  4665  C3  1C  46      10.0          JMP     DDRCEX          ;THEN EXIT
2109
2110                                 ;HERE TO DO A "SCLK" CYCLE
2111
2112  4668  F5                12.0  CLOCK0: PUSH    PSW             ;SAVE A
2113  4669  C5                12.0          PUSH    B               ;SAVE B & C
2114  466A  DB  C0            10.0          IN      DBUSSTA         ;GET CURRENT DBUS STATUS
2115  466C  F6  01            7.0           ORI     M.SCLK          ;ADD IN THE SCLK BIT
2116  466E  47                4.0           MOV     B,A             ;SAVE TEMP RESULT
2117  466F  E6  80            7.0           ANI     M.WREN          ;SEE IF WRITTING OR READING
2118  4671  78                4.0           MOV     A,B             ;RESET REG A
2119  4672  C2  77  46      10.0          JNZ     CLOCK1          ;JUMP OVER IF TM78 IS WRITTING TO HOST
2120  4675  F6  02            7.0           ORI     T.SCLK          ;SET TM78 READ ENABLE
2121  4677  D3  C0            10.0  CLOCK1: OUT     DBUSCTL         ;SET "SCLK"
2122  4679  DB  C0            10.0          IN      DBUSSTA         ;GET MASSBUS STATUS
2123  467B  E6  80            7.0           ANI     M.WREN          ;SAVE ONLY WRITE ENABLE BIT
2124  467D  C2  83  46      10.0          JNZ     CLKSKP          ;SKIP OVER DDR READ CODE IF SET
2125  4680  CD  95  46      18.0          CALL    GETDDR          ;GET DATA TO DDR STORAGE AREA
2126  4683  DB  C0            10.0  CLKSKP: IN      DBUSSTA         ;GET STATUS AGAIN
2127  4685  E6  B0            7.0           ANI     @260            ;SAVE ALL EXCEPT "SCLK"
```

```
2128  4687  47              4.0          MOV   B,A              ;SAVE TEMP RESULT
2129  4688  E6  80          7.0          ANI   M.WREN
2130  468A  78              4.0          MOV   A,B              ;RESET REG A
2131  468B  C2  90  46      10.0          JNZ   CLOCK2           ;JUMP OVER IF WRITTONG TO HOST
2132  468E  F6  02           7.0          ORI   T.SCLK           ;SET READ ENABLE BIT
2133  4690  D3  C0          10.0  CLOCK2: OUT   DBUSCTL         ;CLEAR ''SCLK'' BIT IN DBUS CNTRL WORD
2134  4692  C1              10.0          POP   B                ;RESET THE SAVED REGS
2135  4693  F1              10.0          POP   PSW
2136  4694  C9              10.0          RET                    ;EXIT
2137  4695                               $
 (1)                                     ; *****************************************************************************
2138                                     ;GETDDR -- ROUTINE TO RETRIEVE DATA FROM MASSBUS AND SAVE IT IN THE DDR
2139                                     ;       STORAGE AREA.
2140  4695                               $
 (1)                                     ; *****************************************************************************
2141
2142  4695  DB  D8          10.0  GETDDR: IN    DDRA             ;GET DDRA DATA
2143  4697  32  57  46      13.0          STA   SDDRAT           ;SAVE IT
2144  469A  32  54  46      13.0          STA   SDDRA
2145  469D  DB  D9          10.0          IN    DDRB             ;GET DDRB DATA
2146  469F  32  58  46      13.0          STA   SDDRBT           ;SAVE IT
2147  46A2  32  55  46      13.0          STA   SDDRB
2148  46A5  DB  DA          10.0          IN    DDRC             ;GET DDRC DATA
2149  46A7  32  59  46      13.0          STA   SDDRCT           ;SAVE IT
2150  46AA  32  56  46      13.0          STA   SDDRC
2151  46AD  DB  A1          10.0          IN    CBUSSTA          ;SEE IF ''LEFT'' IS SET
2152  46AF  E6  04           7.0          ANI   W.LEFT           ;SEE IF WMC LEFT IS UP
2153  46B1  C0              12.0          RNZ                    ;RETURN IF SET - NO MODIFICATION NEEDED
2154
2155                                     ;WMC LEFT IS NOT SET (IN RIGHT MODE) MODIFY THE DDR DATA...
2156
2157  46B2  3A  57  46      13.0          LDA   SDDRAT           ;GET TEMP A
2158  46B5  E6  FC           7.0          ANI   $FC              ;STRIP LOW BITS
2159  46B7  0F               4.0          RRC
2160  46B8  0F               4.0          RRC                    ;POSITION TO RIGHT
2161  46B9  47               4.0          MOV   B,A              ;STORE IN REG B
2162  46BA  3A  58  46      13.0          LDA   SDDRBT           ;GET TEMP B
2163  46BD  E6  03           7.0          ANI   $03              ;SAVE ONLY LOW BITS
2164  46BF  0F               4.0          RRC
2165  46C0  0F               4.0          RRC                    ;LEFT JUSTIFY BITS
2166  46C1  B0               4.0          ORA   B                ;COMBINE THE TWO
2167  46C2  32  54  46      13.0          STA   SDDRA            ;SAVE THE DDRA DESIRED
2168
2169  46C5  3A  58  46      13.0          LDA   SDDRBT           ;GET TEMP B AGAIN
2170  46C8  E6  FC           7.0          ANI   $FC              ;SAVE ONLY TOP BITS
2171  46CA  0F               4.0          RRC
2172  46CB  0F               4.0          RRC                    ;RIGHT JUSTIFY
2173  46CC  47               4.0          MOV   B,A              ;TEMP STORE
2174  46CD  3A  59  46      13.0          LDA   SDDRCT           ;GET TEMP DDRC DATA
2175  46D0  E6  03           7.0          ANI   $03              ;SAVE LOW BITS
2176  46D2  0F               4.0          RRC
2177  46D3  0F               4.0          RRC                    ;LEFT JUSTIFY
2178  46D4  B0               4.0          ORA   B                ;COMBINE DATA
2179  46D5  32  55  46      13.0          STA   SDDRB            ;SAVE DESIRED DDRB DATA
```

```
2180
2181  46D8  3A 57 46      13.0          LDA     SDDRAT          ;GET DDRA TEMP DATA
2182  46DB  E6 03          7.0          ANI     $03             ;SAVE LOW BITS
2183  46DD  47              4.0          MOV     B,A             ;TEMP SAVE
2184  46DE  3A 59 46      13.0          LDA     SDDRCT          ;GET DDRC TEMP DATA
2185  46E1  E6 04          7.0          ANI     $04             ;SAVE THE 'P' BIT
2186  46E3  B0              4.0          ORA     B               ;COMBINE DATA
2187  46E4  32 56 46      13.0          STA     SDDRC           ;SAVE DESIRED DDRC DATA
2188  46E7  C9            10.0          RET                     ;EXIT
2189  46E8                          S
 (1)                                  ;*****************************************************************************
2190                                  ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
2191                                  ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
2192                                  ;       THE MASSBUS. WMC 'LEFT' SET INDICATES THAT THIS ROUTINE IS NOT
2193                                  ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
2194                                  ;       'LEFT' IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
2195                                  ;       DDR FORMAT.
2196  46E8                          S
 (1)                                  ;*****************************************************************************
2197
2198  46E8  DB A1          10.0  SETDDR: IN      CBUSSTA         ;GET THE STATUS OF THE WMC 'LEFT' BIT
2199  46EA  E6 04           7.0          ANI     W.LEFT          ;SEE IF THE BIT IS SET
2200  46EC  C2 27 47       10.0          JNZ     SLFTDDR         ;WMC SET TO 'LEFT' MODE - NO TRANSLATION
2201
2202                                  ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR
2203
2204  46EF  3A 57 46      13.0          LDA     SDDRAT          ;GET DDRA TEMP
2205  46F2  E6 3F           7.0          ANI     @77             ;STRIP 2 BITS OFF
2206  46F4  07              4.0          RLC
2207  46F5  07              4.0          RLC                     ;LEFT JUSTIFY
2208  46F6  47              4.0          MOV     B,A             ;TEMP SAVE
2209  46F7  3A 59 46      13.0          LDA     SDDRCT          ;GET DDRC TEMP
2210  46FA  E6 03           7.0          ANI     @3              ;SAVE 2 BITS
2211  46FC  B0              4.0          ORA     B
2212  46FD  32 54 46      13.0          STA     SDDRA           ;SAVE DDRA DESIRED
2213
2214  4700  3A 58 46      13.0          LDA     SDDRBT          ;GET DDRB TEMP
2215  4703  E6 3F           7.0          ANI     @77
2216  4705  07              4.0          RLC
2217  4706  07              4.0          RLC                     ;LEFT JUSTIFY
2218  4707  47              4.0          MOV     B,A
2219  4708  3A 57 46      13.0          LDA     SDDRAT          ;GET DDRA AGAIN
2220  470B  E6 C0           7.0          ANI     @300            ;SAVE 2 BITS
2221  470D  07              4.0          RLC
2222  470E  07              4.0          RLC
2223  470F  B0              4.0          ORA     B
2224  4710  32 55 46      13.0          STA     SDDRB           ;SAVE DDRB DATA
2225
2226  4713  3A 59 46      13.0          LDA     SDDRCT          ;GET DDRC TEMP
2227  4716  E6 04           7.0          ANI     @4              ;SAVE THE PARITY BIT
2228  4718  47              4.0          MOV     B,A             ;TEMP STORE
2229  4719  3A 58 46      13.0          LDA     SDDRBT          ;GET DDRB AGAIN
2230  471C  E6 C0           7.0          ANI     @300            ;SAVE 2 BITS
2231  471E  07              4.0          RLC
```

```
2232  471F  07              4.0          RLC                     ;RIGHT JUSTIFY
2233  4720  B0              4.0          ORA     B
2234  4721  32  56  46      13.0         STA     SDDRC           ;SAVE DDRC DATA
2235  4724  C3  39  47      10.0         JMP     SDDCOM          ;COMMON EXIT
2236                                  ;HERE IF NO MODIFICATION OF DDR DATA NECESSARY
2237
2238  4727  3A  57  46      13.0  SLFTDDR: LDA    SDDRAT          ;GET DDRA DATA
2239  472A  32  54  46      13.0         STA     SDDRA           ;SAVE FOR OUTPUT
2240  472D  3A  58  46      13.0         LDA     SDDRBT          ;GET TEMP DDRB DATA
2241  4730  32  55  46      13.0         STA     SDDRB           ;SAVE IT
2242  4733  3A  59  46      13.0         LDA     SDDRCT          ;GET DDRC TEMP DATA
2243  4736  32  56  46      13.0         STA     SDDRC           ;SAVE FOR OUTPUT
2244
2245  4739  3A  54  46      13.0  SDDCOM: LDA    SDDRA           ;GET DATA TO OUTPUT
2246  473C  D3  D8            10.0         OUT     DDRA
2247  473E  3A  55  46      13.0         LDA     SDDRB
2248  4741  D3  D9            10.0         OUT     DDRB
2249  4743  3A  56  46      13.0         LDA     SDDRC
2250  4746  D3  DA            10.0         OUT     DDRC
2251  4748  C9              10.0         RET                     ;EXIT
2252
2253  4749                          S
 (1)                              ; *****************************************************************************
2254                              ;CAL16P -- CALCULATE 16 BIT PARITY
2255                              ;
2256                              ;       THIS ROUTINE WILL CALCULATE THE CORRECT PARITY
2257                              ;       TO TRANSFER ACROSS THE VAX MASSBUS
2258                              ;       VAX MUST NOT SEND 18 BIT INFORMATION
2259                              ;       ACROSS THE MASSBUS BECAUSE THE VAX DOES NOT
2260                              ;       HAVE 18 DATA LINES.
2261                              ;       THIS SUBROUTINE WILL ALSO CLEAR BITS 16 AND 17
2262                              ;       SO THEY WILL NOT BE SET DURING THE TRANSFER.
2263                              ;       THIS WAS IMPLEMENTED IN THE VAX TM78 LOGIC TEST
2264                              ;       BY CHECKING TO SEE IF THE VAX CHANGED THE FORMAT
2265                              ;       FIELD TO PDP11 NORMAL MODE.
2266                              ;       THE MODE WAS SET BY US TO PDP10 MODE.
2267  4749                          S
 (1)                              ; *****************************************************************************
2268
2269  4749  F5              12.0  CAL16P: PUSH    PSW             ;SAVE A
2270  474A  C5              12.0         PUSH    B
2271  474B                           RIN     R02H            ;READ FORMAT CAS REGISTER
 (1)  474B  DB  85            10.0                 IN      R02H            ;READ R02H INTO AC
 (1)  474D  7F                4.0                  MOV     A,A             ;RETRY LINK
2272  474E  E6  70            7.0          ANI     @160            ;SAVE ONLY FORMAT BITS
2273  4750  C2  80  47      10.0         JNZ     NOTVAX          ;THIS IS NOT A VAX IF NOT ZERO
2274  4753  06  FF            7.0          MVI     B,$FF           ;GET A -1
2275  4755  0E  08            7.0          MVI     C,08            ;SET LOOP COUNTER
2276
2277  4757  3A  57  46      13.0         LDA     SDDRAT          ;GET BYTE A
2278  475A  07              4.0  1$:     RLC                     ;SET 'C' BIT
2279  475B  D2  5F  47      10.0         JNC     2$              ;JUMP IF 'C' BIT NOT SET
2280  475E  04              4.0          INR     B               ;COUNT NUMBER OF BITS
2281  475F  0D              4.0  2$:     DCR     C               ;8 BITS
```

```
2282  4760  C2  5A  47   10.0          JNZ     1$              ;JUMP IF NOT DONE A BYTE
2283
2284  4763  0E  08       7.0          MVI     C,08            ;RESET LOOP COUNTER
2285  4765  3A  58  46  13.0          LDA     SDDRBT          ;GET BYTE B
2286  4768  07            4.0  3$:     RLC                     ;SHIFT LEFT
2287  4769  D2  6D  47   10.0          JNC     4$              ;JUMP IF 'C' BIT NOT SET
2288  476C  04            4.0          INR     B               ;COUNT ANOTHER BIT
2289  476D  0D            4.0  4$:     DCR     C               ;COUNT DOWN 8 BITS
2290  476E  C2  68  47   10.0          JNZ     3$              ;LOOP UNTIL DONE
2291
2292  4771  78            4.0          MOV     A,B             ;COPY NUMBER OF BITS
2293  4772  E6  01       7.0          ANI     001             ;BIT ZERO IS PARITY
2294  4774  CA  7C  47   10.0          JZ      5$              ;'F NOT SET CLEAR PARITY
2295  4777  3E  04       7.0          MVI     A,@004          ;SET PARITY BIT
2296  4779  C3  7D  47   10.0          JMP     6$              ;JUMP
2297
2298  477C  AF            4.0  5$:     XRA     A               ;CLEAR PARITY BIT AND BITS 16,17
2299  477D  32  59  46  13.0  6$:     STA     SDDRCT          ;STORE CORRECT PARITY
2300
2301  4780  C1           10.0  NOTVAX: POP     B
2302  4781  F1           10.0          POP     PSW             ;RESTORE A REGISTER
2303  4782  C9           10.0          RET
2304
2305
2306                                   ;HERE IS THE TABLE OF EXPECTED DATA - 40 18 BIT WORDS
2307
2308  4783  01                   DATATB: .BYTE   1               ;FLOATING 1''S
2309  4784  00                           .BYTE   0
2310  4785  00                           .BYTE   0
2311  4786  02                           .BYTE   2
2312  4787  00                           .BYTE   0
2313  4788  00                           .BYTE   0
2314  4789  04                           .BYTE   4
2315  478A  00                           .BYTE   0
2316  478B  00                           .BYTE   0
2317  478C  08                           .BYTE   @10
2318  478D  00                           .BYTE   0
2319  478E  00                           .BYTE   0
2320  478F  10                           .BYTE   @20
2321  4790  00                           .BYTE   0
2322  4791  00                           .BYTE   0
2323  4792  20                           .BYTE   @40
2324  4793  00                           .BYTE   0
2325  4794  00                           .BYTE   0
2326  4795  40                           .BYTE   @100
2327  4796  00                           .BYTE   0
2328  4797  00                           .BYTE   0
2329  4798  80                           .BYTE   @200
2330  4799  00                           .BYTE   0
2331  479A  00                           .BYTE   0
2332  479B  00                           .BYTE   @0
2333  479C  01                           .BYTE   1
2334  479D  00                           .BYTE   0
2335  479E  00                           .BYTE   0
```

```
2336  479F  02          .BYTE   2
2337  47A0  00          .BYTE   0
2338  47A1  00          .BYTE   0
2339  47A2  04          .BYTE   4
2340  47A3  00          .BYTE   0
2341  47A4  00          .BYTE   0
2342  47A5  08          .BYTE   @10
2343  47A6  00          .BYTE   0
2344  47A7  00          .BYTE   0
2345  47A8  10          .BYTE   @20
2346  47A9  00          .BYTE   0
2347  47AA  00          .BYTE   0
2348  47AB  20          .BYTE   @40
2349  47AC  00          .BYTE   0
2350  47AD  00          .BYTE   0
2351  47AE  40          .BYTE   @100
2352  47AF  00          .BYTE   0
2353  47B0  00          .BYTE   0
2354  47B1  80          .BYTE   @200
2355  47B2  00          .BYTE   0
2356  47B3  00          .BYTE   0
2357  47B4  00          .BYTE   0
2358  47B5  01          .BYTE   1
2359  47B6  00          .BYTE   0
2360  47B7  00          .BYTE   0
2361  47B8  02          .BYTE   2
2362  47B9  FF          .BYTE   @377            ;FLOATING 0'S
2363  47BA  FF          .BYTE   @377
2364  47BB  01          .BYTE   1
2365  47BC  FF          .BYTE   @377
2366  47BD  FF          .BYTE   @377
2367  47BE  02          .BYTE   2
2368  47BF  FF          .BYTE   @377
2369  47C0  7F          .BYTE   @177
2370  47C1  03          .BYTE   3
2371  47C2  FF          .BYTE   @377
2372  47C3  BF          .BYTE   @277
2373  47C4  03          .BYTE   3
2374  47C5  FF          .BYTE   @377
2375  47C6  DF          .BYTE   @337
2376  47C7  03          .BYTE   3
2377  47C8  FF          .BYTE   @377
2378  47C9  EF          .BYTE   @357
2379  47CA  03          .BYTE   3
2380  47CB  FF          .BYTE   @377
2381  47CC  F7          .BYTE   @367
2382  47CD  03          .BYTE   3
2383  47CE  FF          .BYTE   @377
2384  47CF  FB          .BYTE   @373
2385  47D0  03          .BYTE   3
2386  47D1  FF          .BYTE   @377
2387  47D2  FD          .BYTE   @375
2388  47D3  03          .BYTE   3
2389  47D4  FF          .BYTE   @377
```

```
2390  47D5  FE                        .BYTE   @376
2391  47D6  03                        .BYTE   3
2392  47D7  7F                        .BYTE   @177
2393  47D8  FF                        .BYTE   @377
2394  47D9  03                        .BYTE   3
2395  47DA  BF                        .BYTE   @277
2396  47DB  FF                        .BYTE   @377
2397  47DC  03                        .BYTE   3
2398  47DD  DF                        .BYTE   @337
2399  47DE  FF                        .BYTE   @377
2400  47DF  03                        .BYTE   3
2401  47E0  EF                        .BYTE   @357
2402  47E1  FF                        .BYTE   @377
2403  47E2  03                        .BYTE   3
2404  47E3  F7                        .BYTE   @367
2405  47E4  FF                        .BYTE   @377
2406  47E5  03                        .BYTE   3
2407  47E6  FB                        .BYTE   @373
2408  47E7  FF                        .BYTE   @377
2409  47E8  03                        .BYTE   3
2410  47E9  FD                        .BYTE   @375
2411  47EA  FF                        .BYTE   @377
2412  47EB  03                        .BYTE   3
2413  47EC  FE                        .BYTE   @376
2414  47ED  FF                        .BYTE   @377
2415  47EE  03                        .BYTE   3
2416  47EF  FF                        .BYTE   @377            ;ALL 1''S
2417  47F0  FF                        .BYTE   @377
2418  47F1  07                        .BYTE   7
2419  47F2  00                        .BYTE   0               ;ALL 0''S
2420  47F3  00                        .BYTE   0
2421  47F4  04                        .BYTE   4
2422  47F5  55                        .BYTE   @125            ;ALTERNATING BITS
2423  47F6  55                        .BYTE   @125
2424  47F7  01                        .BYTE   1
2425  47F8  AA                        .BYTE   @252            ;COMPLIMENT ALT. BIT DATA
2426  47F9  AA                        .BYTE   @252
2427  47FA  02              DTEND:    .BYTE   2
```

```
                                .SBTTL  TEST 3 - READ DATA TRANSFER USING THE 'DDR' REGISTERS - DDR IN RIGHT MODE
47FB                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*----- -----
                                ;*READ DATA TRANSFER USING THE 'DDR' REGISTERS - DDR IN RIGHT MODE
47FB                            SM
                                ;*********************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8956 & M8959
47FB                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST REQUIRES THE HOST TO READ THE 'MBDPAR' DATA PATTERN WHICH
                                ;*IS 40 18 BIT WORDS LONG. THE BYTE COUNT REGISTER (05) WILL BE LOADED
                                ;*TO 100 BYTES (10 DUMP MODE USED -- 5 BYTES PER WORD * 40 WORDS).
                                ;*
                                ;*EACH 18 BIT WORD WILL BE CHECKED TO THE HOST CPU OVER THE MASSBUS DATA
                                ;*LINES WITH DIAGNOSTIC 'SCLK' SIGNAL.
                                ;*
                                ;*ANY MICROCOMPUTER DETECTED ERROR DETECTED DURING THE DATA TRANSFER WILL
                                ;*BE REPORTED TO THE HOST CPU.
                                ;*
                                ;*READ XFER TEST
47FB                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   INIT TEST-TESTX MACRO
                                ;*   ISSUE WMC RESTART TO INIT SYSTEM
                                ;*   INIT ECODE COUNTER TO 0 + SELECT WMC CONTROL TO 'WRITE' TO HOST CPU
                                ;*   REQUEST HOST TO START A READ FORWARD COMMAND @ 10 DUMP MODE WITH
                                ;*    'MBDPAR' DATA EXPECTED-40 SCLKS OF DATA SENT TO HOST CPU
                                ;*   CONTINUE WHEN HOST RESPONDS WITH A 71 FUNCTION CODE
                                ;*   INIT 'SCLK' COUNT TO '0' + SET MASSBUS 'OCC' + 'RUN' + 'MBWRTEN'
                                ;*   SET 'SCLK' + WAIT SMALL AMOUNT OF TIME
                                ;*   IF HOST RESPOND WITH 'WCLK'?
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-REPORT MASSBUS HANDSHAKE FAILURE ON DATA BUS-OPI ERROR
                                ;*   ENDIF
                                ;*   FINISH CLOCK CYCLE BY CLEARING 'SCLK' IN DBUS CONTROL
                                ;*   IF 'SCLK' GO AWAY?
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-REPORT MASSBUS STATUS ERROR
                                ;*   BGNDO
                                ;*   :  DO UNTIL 40 'SCLKS' OF DATA SENT
                                ;*   :  IF XFR DONE-40 CLKS
                                ;*   :  :  THEN-CONTINUE
                                ;*   :  :  ELSE-
                                ;*   :  ENDIF
                                ;*   ENDDO
```

```
                                        ;*  TERMINATE HOST READ SFR BY SETTING 'EBL'-THEN TERMINATE BY DROPPING
                                        ;*  MASSBUS 'OCC' + 'RUN' BITS IN DBUS CONTROL
                                        ;*  REQUEST HOST CPU TO CHECK THE TERMINATION STATUS FROM READ XFR
                                        ;*  TERMINATE TEST IF HOST RESPONDS WITH A 31-CONTINUE CODE
                                        ;*  OR REPEAT TEST IF HOST RESPONDS WITH A 33-LOOP ON ERROR CODE
                                        ;*ENDTST
47FB                                    SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*MBD1 MICRO TEST 03
                                        ;*MBD1 MICRO ERROR 17
                                        ;*MBD1-DATA READ FROM TM78 TEST USING DIAG DATA REG.-'DDR' IN 'RIGHT' MODE
                                        ;*M8956, M8959
                                        ;*OPI ERROR - HOST CPU FAILED TO SET 'RUN' FOR A READ COMMAND (7X CODE)
                                        ;*TEST ABORTED.
                                        ;*
                                        ;*MBD1 MICRO TEST 03
                                        ;*MBD1 MICRO ERROR 20
                                        ;*MBD1-DATA READ FROM TM78 TEST USING DIAG DATA REG.-'DDR' IN 'RIGHT' MODE
                                        ;*M8956, M8959
                                        ;*'SCLK' STUCK ON ... DIDN'T CLEAR AFTER IT WAS SET FOR DATA TRANSFER
                                        ;*TEST ABORTED!
47FB                                    S
 (1)                                    ; ***************************************************************************

47FB                                    TEST3:  TESTX   @3
47FB  3E  03            7.0                             MVI     A,@3            ;DEFINE THE TEST NUMBER
47FD  CD  03  28       18.0                             CALL    TSET            ;SETUP THE TEST
                                        ;%MBD-DATA READ FROM TM78 TEST USING DIAG. DATA REG.-'DDR' IN 'RIGHT' MODE
                                        ;&M8956, M8959

4800  3E  01             7.0  TST3L:   MVI     A,W.RST         ;RESTART THE WMC
4802  D3  D3            10.0           OUT     WMCCTL

                                        ;FINISH THE WMC RESTART TO CLEAR THE WMC 'LEFT' BIT

4804  AF                4.0           XRA     A               ;NOW DROP WMC 'RST' BIT
4805  D3  D3           10.0           OUT     WMCCTL
4807  3E  B0            7.0           MVI     A,@260          ;INIT THE 'ECODE' COUNTER
4809  D3  D7           10.0           OUT     CNTCTL          ;SELECT THE COUNTER
480B                                   CLRECT
 (1)  480B  AF            4.0                     XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  480C  D3  D6       10.0                     OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  480E  D3  D6       10.0                     OUT     ERRCNT          ;CLEAR BITS 15-8

                                        ;SET THE 'DDR' CONTROL WORD TO 'OUT' TRANSFER

4810  3E  88            7.0           MVI     A,@210
4812  D3  DB           10.0           OUT     DDRCTL          ;INIT THE DDR CONTROL WORD
                                        ;REQUEST THE HOST TO SETUP FOR A 100 BYTE READ TRANSFER (40 18 BIT WORDS)
                                        ;AND EXPECT FUNCTION CODE 71 RETURNED (READ FORWARD COMMAND). PACKING
                                        ;FORMAT IS 10 DUMP MODE.
                                        ;IF THE FORMAT FIELD IS CHANGED TO PDP11 NORMAL
```

```
                                        ;THIS MEANS THAT A VAX HAS ISSUED THE READ SO
                                        ;DON'T CALCULATE THE PARITY ON 18 BITS.
                                        ;THE RH780 ONLY HAS A 16 BIT DATA BUS.

4814  3E 08          7.0   TST3GO: MVI   A,W.WRITE              ;NOW TELL THE WMC TO 'WRITE' SO
4816  D3 D3         10.0           OUT   WMCCTL                 ;DATA REG (DR) WILL NOT BE ENABLED TO MASSBUS
4818                         REQ   3,1,40,@30004
(1) 4818  CD 06 28    18.0                 CALL  REQST
(1) 481B  01                               .BYTE 1              ;DATA PATTERN NUMBER
(1) 481C  00 28                            .WORD 40             ;SYSTEM ''40'' COUNT
(1) 481E  30 04                            .WORD @30004         ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(1) 4820  00                               .BYTE                ;DATA COMPARE FLAG IF =1
(1) 4821  03                               .BYTE 3              ;REQUEST CODE

                                        ;INIT THE SCLK COUNTER TO 1 - TEST WILL QUIT WHEN COUNT IS 41.

4822  3E 01          7.0           MVI   A,1
4824  32 53 46      13.0           STA   CLKCNT                 ;INIT THE CLOCK COUNTER

                                        ;HOST IS READY FOR THE DATA TRANSFER...
                                        ;SET MASSBUS OCCUPIED (OCC), AND 'MB WR EN''

4827  3E A0          7.0           MVI   A,M.WREN.M.OCC
4829  D3 C0         10.0           OUT   DBUSCTL                ;SET OCC + MB WR EN

                                        ;NOW WAIT FOR HOST CPU TO ASSERT 'RUN'' ON THE MASSBUS...COULD TAKE UP TO
                                        ;10 MILLISEC (MASSBUS SPEC.)

482B  0E 14          7.0           MVI   C,20                   ;LOOK FOR 100 MILLI-SEC.
482D  DB C0         10.0   T3W:    IN    DBUSSTA                ;GET DBUS STATUS - CHECK FOR 'RUN''
482F  E6 04          7.0           ANI   M.RUN                  ;SAVE ONLY 'RUN'' BIT
4831  C2 4F 48      10.0           JNZ   TST3WC                 ;JUMP IF RUN IS UP - OK TO PROCEED
4834  3E FF          7.0   T3WL:   MVI   A,@377                 ;GET A DELAY TIME
4836  3D             4.0   T3WL1:  DCR   A                      ;DECREMENT
4837  C2 36 48      10.0           JNZ   T3WL1                  ;STAY IN LOOP TILL -0

                                        ;DOWNCOUNT GROSS TIMER COUNT

483A  0D             4.0           DCR   C
483B  C2 2D 48      10.0           JNZ   T3W                    ;THEN CHECK IF 'RUN'' IS UP
483E  CD E3 48      18.0           CALL  QUITW
4841                               ERR   TSTEND,T3WLC
(1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 4841  CD 09 28    18.0                 CALL  ERLP             ;PROCESS ERROR - DO 2.3
(1)       000F                             MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
(1) 4844  0F                               .BYTE MSGN             ;MESSAGE NUMBER ID
(1) 4845  00                               .BYTE
(1) 4846  CD 15 28    18.0           T3WLC:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
(1) 4849  DA 18 28    10.0                 JC    TSTEND           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>OP! ERROR - HOST CPU FAILED TO SET 'RUN'' FOR A READ COMMAND (7X CODE)
                                        ;<TEST ABORTED!
484C  C3 18 28      10.0           JMP   TSTEND                 ;FATAL ERROR! - QUIT NOW
                                        ;POINT TO THE TABLE OF DATA TO TRANSFER
```

```
2557  484F  21  83  47        10.0  TST3WC: LXI     H,DATATB
2558
2559  4852  7E                 7.0  TST3A:  MOV     A,M             ;GET A BYTE OF DATA
2560  4853  32  57  46         13.0          STA     SDDRAT          ;SAVE DDR REG A - BITS 7-0
2561  4856  23                 6.0          INX     H               ;POINT TO NEXT DATA BYTE
2562  4857  7E                 7.0          MOV     A,M             ;GET THE DATA
2563  4858  32  58  46         13.0          STA     SDDRBT          ;SAVE IN DDR REG B - BITF 15-8
2564  485B  23                 6.0          INX     H               ;POINT TO THE NEXT DATA BYTE
2565  485C  7E                 7.0          MOV     A,M             ;GET IT
2566  485D  32  59  46         13.0          STA     SDDRCT          ;SAVE IN DDR REG C - BITS "P" + 17-16
2567  4860  CD  49  47         18.0          CALL    CAL16P          ;SEE IF THIS IS A VAX AND CORRECT
2568                                                                ;DATA IF SO
2569  4863  CD  E8  46         18.0          CALL    SETDDR          ;LOAD THE DDR REGS
2570                                  ;START THE DATA TRANSFER BY 1ST SETTING "SCLK"
2571
2572  4866  3E  A1               7.0  TST3B:  MVI     A,M.WREN.M.OCC.M.SCLK  ;SET DBUS CNTRL WORD TO
2573  4868  D3  C0              10.0          OUT     DBUSCTL         ;"MB WR EN" + "OCC" + "SCLK"
2574
2575                                  ;FINISH THE 1ST SCLK CYCLE BY DROPPING THE SCLK BIT IN DBUS CNTRL WORD
2576
2577  486A  3E  A0               7.0  TST3BB: MVI     A,M.WREN!M.OCC  ;SET DBUS CNTRL WORD TO
2578  486C  D3  C0              10.0          OUT     DBUSCTL         ;"MB WR EN" + "OCC"
2579  486E  00                   4.0          NOP                     ;SMALL DELAY
2580  486F  DB  C0              10.0          IN      DBUSSTA         ;GET DBUS CONTROL STATUS
2581  4871  E6  01               7.0          ANI     M.SCLK          ;SAVE ONLY SCLK BIT
2582  4873  CA  87  48         10.0          JZ      TST3C           ;JUMP OVER IF THE BIT CLEARED AS EXPECTED
2583  4876  CD  E3  48         18.0          CALL    QUITW           ;ABORT THE WRITE XFR
2584  4879                                  ERR     TST3L,TST3BC
  (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4879  CD  09  28         18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)       0010                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  487C  10                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  487D  00                           .BYTE
  (1)  487E  CD  15  28         18.0  TST3BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4881  DA  00  48         10.0                  JC      TST3L           ;LOOP ADDRESS IF LOOP SPECIFIED
2585                                  ;>"SCLK" STUCK ON ... DIDN'T CLEAR AFTER IT WAS SET FOR DATA TRANSFER
2586                                  ;<TEST ABORTED!
2587  4884  C3  18  28         10.0          JMP     TSTEND          ;QUIT TESTING
2588
2589                                  ;SETUP FOR THE NEXT DATA TRANSFER WORD
2590  4887  23                 6.0  TST3C:  INX     H               ;POINT TO NEXT DATA BYTE WORD
2591  4888  7E                 7.0          MOV     A,M             ;GET A BYTE OF DATA
2592  4889  32  57  46         13.0          STA     SDDRAT          ;SAVE DDR REG A - BITS 7-0
2593  488C  23                 6.0          INX     H               ;POINT TO NEXT DATA BYTE
2594  488D  7E                 7.0          MOV     A,M             ;GET THE DATA
2595  488E  32  58  46         13.0          STA     SDDRBT          ;SAVE IN DDR REG B - BITE 15-8
2596  4891  23                 6.0          INX     H               ;POINT TO THE NEXT DATA BYTE
2597  4892  7E                 7.0          MOV     A,M             ;GET IT
2598  4893  32  59  46         13.0          STA     SDDRCT          ;SAVE IN DDR REG C - BITS "P" + 17-16
2599  4896  CD  49  47         18.0          CALL    CAL16P          ;SEE IF THIS IS A VAX HOST AND CORRECT
2600                                                                ;DATA IF SO
2601  4899  CD  E8  46         18.0          CALL    SETDDR          ;LOAD THE DDR REGS
2602
2603                                  ;COUNT THE SCLK CYCLE (18 BIT WORD TRANSFERED)
```

```
2604
2605  489C  3A  53  46   13.0          LDA     CLKCNT          ;GET CURRENT CLOCK COUNT
2606  489F  3C                 4.0     INR     A               ;UPDATE IT
2607  48A0  32  53  46   13.0          STA     CLKCNT          ;SAVE UPDATED COUNT
2608  48A3  FE  29               7.0    CPI     41              ;DONE WITH TRANSFER?
2609  48A5  CA  AE  48   10.0          JZ      T3END           ;YES
2610
2611  48A8  CD  68  46   18.0          CALL    CLOCKO          ;NO - CLOCK THE DATA LOADED IN DDR TO HOST
2612  48AB  C3  87  48   10.0          JMP     TST3C           ;AND LOOP BACK FOR THE NEXT WORD
2613                        ;HERE WHEN THE TEST IS FINISHED WITH THE DATA TRANSFER - TERMINATE
2614                        ;THE TRANSFER AND GIVE UP THE MASSBUS DATA LINES
2615
2616  48AE  3E  B0               7.0  T3END:  MVI     A,M.WREN!M.OCC!M.EBL  ;SET DBUS CNTRL WORD TO
2617  48B0  D3  C0              10.0          OUT     DBUSCTL         ;''MB WR EN'' + ''OCC'' + ''EBL''
2618  48B2  3E  A0               7.0          MVI     A,M.WREN!M.OCC  ;DROP ''EBL'' SIGNAL TO MASSBUS
2619  48B4  D3  C0              10.0          OUT     DBUSCTL
2620  48B6  3E  00               7.0          MVI     A,0
2621  48B8  D3  C0              10.0          OUT     DBUSCTL         ;THEN RELEASE THE MASSBUS
2622
2623                        ;REQUEST THE HOST CPU TO CHECK THE DATA JUST TRANSFERED TO HOST MEMORY
2624                        ;AND TO CHECK THE TERMINATION STATUS OF THE DATA TRANSFER FROM THE HOST
2625                        ;CPU SIDE OF THE MASSBUS. HOST CPU WILL REPORT ANY DETECTED ERROR AND
2626                        ;RESPOND WITH CONTINUE CODE (31) TO TERMINATE THIS TEST OR WITH THE
2627                        ;LOOP-ON-ERROR CODE (33) TO ISSUE THE TEST AGAIN.
2628
2629  48BA                          REQ     5,1,40,@30004,1
 (1)  48BA  CD  06  28   18.0                       CALL    REQST
 (1)  48BD  01                                     .BYTE   1               ;DATA PATTERN NUMBER
 (1)  48BE  00  28                                 .WORD   40              ;SYSTEM ''40'' COUNT
 (1)  48C0  30  04                                 .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  48C2  01                                     .BYTE   1               ;DATA COMPARE FLAG IF -1
 (1)  48C3  05                                     .BYTE   5               ;REQUEST CODE
2630                        ;CHECK WHAT RESPONSE CODE WAS RECEIVED BY THE MICROCOMPUTER...
2631                        ;ISSUE THE TEST AGAIN IF CODE 33 RECEIVED.
2632
2633  48C4  3A  98  4F   13.0          LDA     VALFC           ;GET THE FUNCTION CODE RECEIVED
2634  48C7  FE  1B               7.0   CPI     @33             ;LOOP-ON-ERROR CODE?
2635  48C9  CA  00  48   10.0          JZ      TST3L           ;YES - DO TEST AGAIN
2636  48CC                          ENDTST  TST3L
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  48CC                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  48CC  CD  06  28   18.0                       CALL    REQST
 (2)  48CF  00                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  48D0  00  00                                 .WORD                   ;SYSTEM ''0'' COUNT
 (2)  48D2  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  48D4  00                                     .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  48D5  07                                     .BYTE   7               ;REQUEST CODE
 (1)  48D6  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  48D9  3D               4.0       DCR     A               ;DOWNCOUNT
 (1)  48DA  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  48DD  F2  00  48   10.0          JP      TST3L           ;DO TEST UNTIL TILL = 0
2637  48E0  C3  F0  48   10.0          JMP     TEST4           ;DO THE NEXT TEST
2638                        ;HERE IS AN ERROR WAS DETECTED ON THE SCLK CYCLE - TERMINATE THE DATA
2639                        ;TRANSFER BY SENDING MASSBUS AN EBL AND EXC SIGNAL
```

```
2640  48E3  3E  88          7.0  QUITW:  MVI   A,M.WREN!M.EXC  ;SET DBUS CNTRL WORD TO
2641  48E5  D3  C0         10.0          OUT   DBUSCTL         ;SET EXC
2642  48E7  3E  18          7.0          MVI   A,M.EBL.M.EXC   ;SET BOTH EXC AND EBL
2643  48E9  D3  C0         10.0          OUT   DBUSCTL
2644  48EB  3E  00          7.0          MVI   A,0
2645  48ED  D3  C0         10.0          OUT   DBUSCTL         ;RELEASE THE MASSBUS
2646  48EF  C9              10.0          RET                   ;EXIT
```

```
                                .SBTTL  TEST 4 - READ DATA TRANSFER USING THE 'DDR' REGISTERS - DDR IN 'LEFT' MODE
 2649  48F0                     ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*----- -----
                                ;*READ DATA TRANSFER USING THE 'DDR' REGISTERS - DDR IN 'LEFT' MODE
 2651  48F0                     SM
                                ;*********************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8956 & M8959
 2653  48F0                     SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST REQUIRES THE HOST TO READ THE 'MBDPAR' DATA PATTERN WHICH
                                ;*IS 40 18 BIT WORDS LONG. THE BYTE COUNT REGISTER (05) WILL BE LOADED
                                ;*TO 100 BYTES (10 DUMP MODE USED -- 5 BYTES PER WORD * 40 WORDS).
                                ;*
                                ;*EACH 18 BIT WORD WILL BE CHECKED TO THE HOST CPU OVER THE MASSBUS DATA
                                ;*LINES WITH DIAGNOSTIC "SCLK" SIGNAL.
                                ;*
                                ;*ANY MICROCOMPUTER DETECTED ERROR DETECTED DURING THE DATA TRANSFER WILL
                                ;*BE REPORTED TO THE HOST CPU.
                                ;*
                                ;*READ XFER TEST
 2665  48F0                     SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  INIT TEST-TESTX MACRO
                                ;*  ISSUE WMC RESTART TO INIT SYSTEM
                                ;*  SET WMC IN 'LEFT' MODE
                                ;*  IF WMC 'LEFT' IS SET
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT ERROR
                                ;*  ENDIF
                                ;*  INIT ECODE COUNTER TO 0 + SELECT WMC CONTROL TO 'WRITE' TO HOST CPU
                                ;*  REQUEST HOST TO START A READ FORWARD COMMAND @ 10 DUMP MODE WITH
                                ;*   'MBDPAR' DATA EXPECTED-40 SCLKS OF DATA SENT TO HOST CPU
                                ;*  CONTINUE WHEN HOST RESPONDS WITH A 71 FUNCTION CODE
                                ;*  INIT 'SCLK' COUNT TO '0' + SET MASSBUS 'OCC' + 'RUN' + 'MBWRTEN'
                                ;*  SET 'SCLK' + WAIT SMALL AMOUNT OF TIME
                                ;*  IF HOST RESPOND WITH 'WCLK'?
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT MASSBUS HANDSHAKE FAILURE ON DATA BUS-OPI ERROR
                                ;*  ENDIF
                                ;*  FINISH CLOCK CYCLE BY CLEARING 'SCLK' IN DBUS CONTROL
                                ;*  IF 'SCLK' GO AWAY?
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT MASSBUS STATUS ERROR
                                ;*  BGNDO
                                ;*  :  DO UNTIL 40 'SCLKS' OF DATA SENT
```

```
                                ;*  :  IF XFR DONE-40 CLKS
                                ;*  :  :  THEN-CONTINUE
                                ;*  :  :  ELSE-
                                ;*  :  ENDIF
                                ;*  ENDDO
                                ;*  TERMINATE HOST READ SFR BY SETTING 'EBL'-THEN TERMINATE BY DROPPING
                                ;*  MASSBUS 'OCC' + 'RUN' BITS IN DBUS CONTROL
                                ;*  REQUEST HOST CPU TO CHECK THE TERMINATION STATUS FROM READ XFR
                                ;*  TERMINATE TEST IF HOST RESPONDS WITH A 31-CONTINUE CODE
                                ;*  OR REPEAT TEST IF HOST RESPONDS WITH A 33-LOOP ON ERROR CODE
                                ;*ENDTST
48F0                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MBD1 MICRO TEST 04
                                ;*MBD1 MICRO ERROR 21
                                ;*MBD1-READ FROM TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
                                ;*M8956, M8959
                                ;*WMC FAILED TO SET "WMC LEFT" STATUS BIT IN DDR CONTROL BYTE
                                ;*ACTUAL = NNNN
                                ;*TEST ABORTED!
                                ;*
                                ;*MBD1 MICRO TEST 04
                                ;*MBD1 MICRO ERROR 22
                                ;*MBD1-READ FROM TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
                                ;*M8956, M8959
                                ;*OPI ERROR - HOST CPU FAILED TO SET "RUN" FOR A READ COMMAND (7X CODE)
                                ;*TEST ABORTED!
48F0                            S
 (1)                            ; ************************************************************************

                                TEST4:  TESTX   @4
 (1) 48F0  3E  04        7.0                    MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1) 48F2  CD  03  28    18.0                    CALL    TSET            ;SETUP THE TEST
                                ;%MBD1-READ FROM TM78 TEST USING DIAG. DATA REG.-"DDR" IN "LEFT" MODE
                                ;&M8956, M8959

48F5  3E  B0             7.0    TST4L:  MVI     A,@260          ;INIT THE "ECODE" COUNTER
48F7  D3  D7            10.0            OUT     CNTCTL          ;SELECT THE COUNTER
48F9                                    CLRECT
 (1) 48F9  AF             4.0                    XRA     A               ;CLEAR THE ACCUMULATOR
 (1) 48FA  D3  D6        10.0                    OUT     ERRCNT          ;CLEAR BITS 7-0
 (1) 48FC  D3  D6        10.0                    OUT     ERRCNT          ;CLEAR BITS 15-8

                                ;SET THE "DDR" CONTROL WORD TO "OUT" TRANSFER

48FE  3E  88             7.0            MVI     A,@210
4900  D3  DB            10.0            OUT     DDRCTL          ;INIT THE DDR CONTROL WORD

                                ;TRY TO SET WRITE MICRO "WMC LEFT" BIT

4902  3E  39             7.0            MVI     A,@71           ;FMT=3 (10 DUMP), SKIP CNT = 11 (SET WMC LEFT)
4904  D3  D0            10.0            OUT     DATACTL         ;GO....
```

```
2735  4906  3E  01          7.0          MVI   A,W.RST          ;RESTART THE WMC
2736  4908  D3  D3         10.0          OUT   WMCCTL
2737  490A  AF              4.0          XRA   A                ;NOW DROP WMC "RST" BIT
2738  490B  D3  D3         10.0          OUT   WMCCTL
2739  490D  00              4.0          NOP                    ;SMALL DELAY
2740  490E  3E  80          7.0          MVI   A,W.ENAB         ;TELL WMC TO SET THE LEFT BIT
2741  4910  D3  D3         10.0          OUT   WMCCTL           ;TRY NOW!
2742
2743  4912  3E  28          7.0          MVI   A,40             ;WAIT FOR ABOUT 200 MICRO-SECONDS
2744  4914  3D              4.0  T4WS:   DCR   A                ;DECREMENT THE COUNT
2745  4915  C2  14  49      10.0          JNZ   T4WS             ;WAIT A WHILE...
2746                                   ;NOW CHECK THAT THE WMC PGM REALLY SET "WMC LEFT" BIT SO LOADING DDRA-DDRC
2747                                   ;WILL BE DIRECT TRANSLATION TO MASSBUS BITS.
2748
2749  4918  DB  A1         10.0          IN    CBUSSTA          ;GET CBUS STATUS (WMC LEFT BIT)
2750  491A                               ROUT  ADATA            ;SAVE ACTJAL STATUS FOR ERROR REPORT
 (1)  491A  D3  94         10.0                OUT   ADATA              ;WRITE AC INTO ADATA
 (1)  491C  7F              4.0                MOV   A,A                ;RETRY LINK
2751  491D  E6  04          7.0          ANI   W.LEFT           ;SAVE ONLY WMC LEFT BIT
2752  491F  C2  30  49      10.0          JNZ   TST4GO           ;CONTINUE IF ITS SET...
2753  4922                               ERRA  TST4L,TST4LC
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4922  CD  0F  28      18.0                CALL  ERLPA              ;PROCESS ERROR - DO 2.3
 (1)        0011                              MSGN  =     MSGN+1             ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4925  11                                .BYTE MSGN               ;MESSAGE NUMBER ID
 (1)  4926  00                                .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  4927  CD  15  28      18.0          TST4LC::    CALL  CKLOP              ;CHECK LOOP FUNCTION - DO 2.2
 (1)  492A  DA  F5  48      10.0                JC    TST4L              ;LOOP ADDRESS IF LOOP SPECIFIED
2754                                   ;>WMC FAILED TO SET "WMC LEFT" STATUS BIT IN DDR CONTROL BYTE
2755                                   ;<TEST ABORTED!
2756  492D  C3  18  28      10.0          JMP   TSTEND           ;ABORT RIGHT NOW!
2757                                   ;REQUEST THE HOST TO SETUP FOR A 100 BYTE READ TRANSFER (40 18 BIT WORDS)
2758                                   ;AND EXPECT FUNCTION CODE 71 RETURNED (READ FORWARD COMMAND). PACKING
2759                                   ;FORMAT IS 10 DUMP MODE.
2760                                   ;IF THE FORMAT FIELD IS CHANGED TO PDP11 NORMAL
2761                                   ;THIS MEANS THAT A VAX HAS ISSUED THE READ SO
2762                                   ;DON'T CALCULATE THE PARITY ON 18 BITS.
2763                                   ;THE RH780 ONLY HAS A 16 BIT DATA BUS
2764
2765  4930  3E  08          7.0  TST4GO: MVI   A,W.WRITE                ;NOW TELL THE WMC TO "WRITE" SO
2766  4932  D3  D3         10.0          OUT   WMCCTL                   ;DATA REG (DR) WILL NOT BE ENABLED TO MASSBUS
2767  4934                               REQ   3,1,40,@30004
 (1)  4934  CD  06  28      18.0                CALL  REQST
 (1)  4937  01                                .BYTE 1                  ;DATA PATTERN NUMBER
 (1)  4938  00  28                            .WORD 40                 ;SYSTEM "40" COUNT
 (1)  493A  30  04                            .WORD @30004             ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  493C  00                                .BYTE                    ;DATA COMPARE FLAG IF =1
 (1)  493D  03                                .BYTE 3                  ;REQUEST CODE
2768
2769                                   ;INIT THE SCLK COUNTER TO 0 - TEST WILL QUIT WHEN COUNT IS 41.
2770
2771  493E  3E  00          7.0          MVI   A,0              ;SET CLOCK TO "0" TICK
2772  4940  32  53  46      13.0          STA   CLKCNT           ;INIT THE CLOCK COUNTER
2773
```

```
2774                                    ;HOST IS READY FOR THE DATA TRANSFER...
2775                                    ;SET MASSBUS OCCUPIED (OCC), MASSBUS RUN (RUN), AND 'MB WR EN'
2776
2777  4943  3E  A0          7.0         MVI     A,M.WREN!M.OCC
2778  4945  D3  C0         10.0         OUT     DBUSCTL         ;SET OCC + RUN + MB WR EN
2779
2780                                    ;NOW WAIT FOR HOST CPU TO ASSERT 'RUN' ON THE MASSBUS...COULD TAKE UP TO
2781                                    ;10 MILLISEC (MASSBUS SPEC.)
2782
2783  4947  0E  14          7.0         MVI     C,20            ;LOOK FOR 100 MILLI-SEC.
2784  4949  DB  C0         10.0  T4W:    IN      DBUSSTA         ;GET DBUS STATUS - CHECK FOR 'RUN'
2785  494B  E6  04          7.0         ANI     M.RUN           ;SAVE ONLY 'RUN' BIT
2786  494D  C2  6B  49      10.0         JNZ     TST4WC          ;JUMP IF RUN IS UP - OK TO PROCEED
2787  4950  3E  FF          7.0  T4WL:   MVI     A,@377          ;GET A DELAY TIME
2788  4952  3D              4.0  T4WL1:  DCR     A               ;DECREMENT
2789  4953  C2  52  49      10.0         JNZ     T4WL1           ;STAY IN LOOP TILL =0
2790
2791                                    ;DOWNCOUNT GROSS TIMER COUNT
2792
2793  4956  0D              4.0         DCR     C
2794  4957  C2  49  49      10.0         JNZ     T4W             ;THEN CHECK IF 'RUN' IS UP
2795  495A  CD  E3  48      18.0         CALL    QUITW
2796  495D                               ERR     TSTEND,T4WLC
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  495D  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0012                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4960  12                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4961  00                           .BYTE
 (1)  4962  CD  15  28      18.0  T4WLC:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4965  DA  18  28      10.0         JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2797                                    ;>OPI ERROR - HOST CPU FAILED TO SET 'RUN' FOR A READ COMMAND (7X CODE)
2798                                    ;<TEST ABORTED!
2799  4968  C3  18  28      10.0         JMP     TSTEND          ;FATAL ERROR! - QUIT NOW
2800                                    ;POINT TO THE TABLE OF DATA TO TRANSFER
2801
2802  496B  21  82  47      10.0  TST4WC: LXI     H,DATATB-1      ;SETUP THE POINTER
2803
2804                                    ;SETUP FOR THE NEXT DATA TRANSFER WORD
2805
2806  496E  23              6.0  TST4C:  INX     H               ;POINT TO NEXT DATA BYTE WORD
2807  496F  7E              7.0         MOV     A,M             ;GET A BYTE OF DATA
2808  4970  32  57  46      13.0         STA     SDDRAT          ;SAVE DDR REG A - BITS 7-0
2809  4973  23              6.0         INX     H               ;POINT TO NEXT DATA BYTE
2810  4974  7E              7.0         MOV     A,M             ;GET THE DATA
2811  4975  32  58  46      13.0         STA     SDDRBT          ;SAVE IN DDR REG B - BITE 15-8
2812  4978  23              6.0         INX     H               ;POINT TO THE NEXT DATA BYTE
2813  4979  7E              7.0         MOV     A,M             ;GET IT
2814  497A  32  59  46      13.0         STA     SDDRCT          ;SAVE IN DDR REG C - BITS 'P' + 17-16
2815  497D  CD  49  47      18.0         CALL    CAL16P          ;SEE IF THE HOST IS A VAX AND
2816                                                            ;CORRECT DATA IF SO
2817  4980  CD  E8  46      18.0         CALL    SETDDR          ;LOAD THE DDR REGS
2818
2819                                    ;COUNT THE SCLK CYCLE (18 BIT WORD TRANSFERED)
2820
```

```
2821  4983  3A  53  46  13.0          LDA     CLKCNT          ;GET CURRENT CLOCK COUNT
2822  4986  3C                4.0     INR     A               ;UPDATE IT
2823  4987  32  53  46  13.0          STA     CLKCNT          ;SAVE UPDATED COUNT
2824  498A  FE  29                7.0     CPI     41          ;DONE WITH TRANSFER? (STOP AFTER 40 SCLKS)
2825  498C  CA  95  49  10.0          JZ      T4END           ;YES
2826
2827  498F  CD  68  46  18.0          CALL    CLOCK0          ;NO - CLOCK THE DATA LOADED IN DDR TO HOST
2828  4992  C3  6E  49  10.0          JMP     TST4C           ;AND LOOP BACK FOR THE NEXT WORD
2829                                ;HERE WHEN THE TEST IS FINISHED WITH THE DATA TRANSFER - TERMINATE
2830                                ;THE TRANSFER AND GIVE UP THE MASSBUS DATA LINES
2831
2832  4995  3E  B0                7.0  T4END: MVI     A,M.WREN.M.OCC!M.EBL  ;SET DBUS CNTRL WORD TO
2833  4997  D3  C0               10.0          OUT     DBUSCTL         ;"MB WR EN" + "OCC" + "EBL"
2834  4999  3E  A0                7.0          MVI     A,M.WREN.M.OCC  ;DROP "EBL" SIGNAL TO MASSBUS
2835  499B  D3  C0               10.0          OUT     DBUSCTL
2836  499D  3E  00                7.0          MVI     A,0
2837  499F  D3  C0               10.0          OUT     DBUSCTL         ;THEN RELEASE THE MASSBUS
2838
2839                                ;REQUEST THE HOST CPU TO CHECK THE DATA JUST TRANSFERED TO HOST MEMORY
2840                                ;AND TO CHECK THE TERMINATION STATUS OF THE DATA TRANSFER FROM THE HOST
2841                                ;CPU SIDE OF THE MASSBUS. HOST CPU WILL REPORT ANY DETECTED ERROR AND
2842                                ;RESPOND WITH CONTINUE CODE (31) TO TERMINATE THIS TEST OR WITH THE
2843                                ;LOOP-ON-ERROR CODE (33) TO ISSUE THE TEST AGAIN.
2844
2845  49A1                          REQ     5,1,40,@30004,1
 (1)  49A1  CD  06  28  18.0                        CALL    REQST
 (1)  49A4  01                                      .BYTE   1               ;DATA PATTERN NUMBER
 (1)  49A5  00  28                                  .WORD   40              ;SYSTEM "40" COUNT
 (1)  49A7  30  04                                  .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  49A9  01                                      .BYTE   1               ;DATA COMPARE FLAG IF -1
 (1)  49AA  05                                      .BYTE   5               ;REQUEST CODE
2846
2847                                ;CHECK WHAT RESPONSE CODE WAS RECEIVED BY THE MICROCOMPUTER...
2848                                ;ISSUE THE TEST AGAIN IF CODE 33 RECEIVED.
2849
2850  49AB  3A  98  4F  13.0          LDA     VALFC           ;GET THE FUNCTION CODE RECEIVED
2851  49AE  FE  1B                7.0     CPI     @33         ;LOOP-ON-ERROR CODE?
2852  49B0  CA  F5  48  10.0          JZ      TST4L           ;YES - DO TEST AGAIN
2853  49B3                            ENDTST  TST4L
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  49B3                      REQ     7                   ;FAKE CALL TO KEEP TEST ALIVE
 (2)  49B3  CD  06  28  18.0                        CALL    REQST
 (2)  49B6  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  49B7  00  00                                  .WORD                   ;SYSTEM "" COUNT
 (2)  49B9  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  49BB  00                                      .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  49BC  07                                      .BYTE   7               ;REQUEST CODE
 (1)  49BD  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  49C0  3D                4.0     DCR     A               ;DOWNCOUNT
 (1)  49C1  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  49C4  F2  F5  48  10.0          JP      TST4L           ;DO TEST UNTIL TILL = 0
2854  49C7  C3  18  28  10.0          JMP     TSTEND          ;EXIT - ALL DONE.
2855
2856        0000                .END
```

```
A      =%0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
ARAIDF= 0098    ASAV     4651   ASAVE    4F9B   ATTCD    4F97
AXNUM    4F91   B      =%0000   BADST  = 0090   BIT0   = 0001
BIT1   = 0002   BIT15 = 8000   BIT2   = 0004   BIT3   = 0008
BIT4   = 0010   BIT5   = 0020   BIT6   = 0040   BIT7   = 0080
BIT8   = 0100   BIT9   = 0200   BRKPBC= 4F0A   BRKRAM= 4F10
BRKSTR= 4E60    BRKXCT= 4F00    BSAVE    4F9C   BYTCNT= 00D4
BYTEH    4F24   BYTEL    4F23   C      =%0001   CAL16P   4749
CASCT    4F21   CASCTL= 00A0    CASSTA= 00A0    CATTH  = 0089
CATTL  = 0088   CBUSST= 00A1    CBYTH  = 008B   CBYTL  = 008A
CDG1H  = 0087   CDG1L  = 0086   CDG2H  = 0093   CDG2L  = 0092
CDG3H  = 0095   CDG3L  = 0094   CDVTH  = 008D   CDVTL  = 008C
CHPTIE= 0028    CHOTIE= 0020    CH1TIE= 0021    CH2TIE= 0022
CH3TIE= 0023    CH4TIE= 0024    CH5TIE= 0025    CH6TIE= 0026
CH7TIE= 0027    CKLOP  = 2815   CLKCNT   4653   CLKCTL= 00F0
CLKSKP   4683   CLOCK    4F26   CLOCK0   4668   CLOCK1   4677
CLOCK2   4690   CMCOH  = 0099   CMCOL  = 0098   CMC1H  = 009B
CMC1L  = 009A   CMC2H  = 009D   CMC2L  = 009C   CMC3H  = 009F
CMC3L  = 009E   CMINH  = 0097   CMINL  = 0096   CNTCTL= 00D7
COMCK    45BA   CRCWRD= 0018    CSAVE    4F9D   CSRLH  = 0091
CSRLL  = 0090   CTCH   = 0085   CTCL   = 0084   CTSTH  = 008F
CTSTL  = 008E   CXCTH  = 0081   CXCTL  = 0080   CXINH  = 0083
CXINL  = 0082   C.     = 0001   C.AVAI= 0080   C.DP   = 0008
C.DSE  = 0010   C.DTU  = 0003   C.DVA  = 0008   C.FAIL= 00FC
C.FMT  = 0070   C.FNCT= 003E    C.GO   = 0001   C.INTC= 00FE
C.MAIN= 0020    C.NSA  = 0080   C.RCT  = 00FC   C.SER  = 0080
C.SHR  = 0040   C.SKPC= 000F    C.TAPE= 0040    C.WCS  = 0002
D      -%0002   DATACT= 00D0    DATATB   4783   DBUS     4F28
DBUSCT= 00C0    DBUSST= 00C0    DDRA   = 00D8   DDRACK   45AC
DDRAIN= 0010    DDRB   = 00D9   DDRBCK   45C3   DDRBIN= 0002
DDRC   = 00DA   DDRCCK   45FD   DDRCED   4621   DDRCER   465D
DDRCEX   461C   DDRCIN= 0001    DDRCO  = 0088   DDRCTL= 00DB
DIAFLG   4F22   DIAGPG= 4300    DIAGRM= 4F90    DIARD  - 000B
DONE1  = 0045   DONINT= 0010    DSAVE    4F9E   DSE    = 0006
DTEND    47FA   D.ATH0= 0001    D.ATH1= 0002    D.EOTD= 0010
D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004    D.TACH= 0008
D.WR4  = 0080   E      =%0003   ECCBAD= 0042    ECCCOR= 0019
ECCOK  = 0041   ECCSTA= 001A    ECCTST= 000E    EDATA  = 0095
EOTCLR= 0003    ERFLG    4F93   ERLP   = 2809   ERLPA  = 280F
ERLPB  = 2812   ERLPE  = 280C   ERNUM    4F90   ERRCNT= 00D6
ESAVE    4F9F   E.ACRC= 0010    E.AMT  = 0020   E.CDP  = 0080
E.CRC  = 0080   E.PNTR= 0008    E.RPE  = 0040   E.STEC= 0001
E.TTEC= 0002    E.UNC  = 0004   FIFORD= 006A    FORMAT   4F25
FWDTST= 0061    GCRID  = 0089   GCRSET= 0002   GETDDR   4695
GOODTM= 0092    H      =%0004   HLSAVE   4FA0   IE     = 0008
INTSTA= 00E0    ITERA    4F9A   I.PWR  = 0020   I.RMPE= 0040
I5.5   = 0010   I6.5   = 0020   I7.5    = 0040   KCALL  = 005F
KCLR   = 007B   KDEP   = 003F   KENAB  = 0078   KEXAM  = 003E
KEYBRD= 00C8    KEY1   = 0078   KEY10  = 006D   KEY11  = 006E
KEY12  = 006F   KEY13  = 005C   KEY14  - 005D   KEY15  = 005E
KEY16  = 005F   KEY17  = 003C   KEY18  = 003D   KEY19  = 003E
KEY2   = 0079   KEY20  = 003F   KEY3   = 007A   KEY4   = 007B
KEY5   = 0074   KEY6   = 0075   KEY7   = 0076   KEY8   = 0077
KEY9   - 006C   KINTA  - 006F   KLDAD  - 003D   KNO    = 003C
```

```
KN1    = 005C          KN2    = 005D          KN3    = 005E          KN4    = 006C
KN5    = 006D          KN6    = 006E          KN7    = 0074          KN8    = 0075
KN9    = 0076          KU2    = 0079          KU3    = 007A          KU8    = 0077
L      =%0005          LBLANK= 000F           LCE    = 000B          LCH    = 000C
LCL    = 000D          LCP    = 000E          LC0    = 0000          LC1    = 0001
LC2    = 0002          LC3    = 0003          LC4    = 0004          LC5    = 0005
LC6    = 0006          LC7    = 0007          LC8    = 0008          LC9    = 0009
LDLEDA= 00CA           LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF           LD18BI  45D4           LKDIAG= 2800
LKKBD  = 004C          LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055
LKLWPG= 0052           LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046
LPFLG   4F94           LPNUM   4F92           M      =%0006          MBSEL  = 00E0
MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF           MINUS  = 000A
MM     = 8000          MSE    = 0008          MSGN   = 0012          MTACLR= 0000
MT.ARA= 0020           MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040
MT.INH= 0008           MT.LWR= 0004           MT.MOT= 0002           MT.NWT= 0080
MT.PEC= 0040           MT.PSB= 0004           MT.PS0= 0001           MT.PS1= 0002
MT.REV= 0020           MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080
M.CAPE= 0020           M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010
M.EXC  = 0008          M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010
M.OCC  = 0020          M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080
M.RDEN= 0002           M.RDPE= 0008           M.RUN  = 0004          M.SCLK= 0001
M.TRA  = 0040          M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080
M.WREN= 0080           M5.5   = 0001          M6.5   = 0002          M7.5   = 0004
NOTCAP= 0088           NOTVAX  4780           OKAY   = 00FF          OPRRAM= 4300
OPSTRT= 0058           OPVER  = 0040          PADCNT= 00D5           PADCRC= 0080
PBIT    4652           PDIAG  = 0048          PEID   = 008A          PENAB  = 004C
PESET  = 0001          PL     = 00B1          PNMOD   4643           PRDD   = 004C
PRENF  = 009C          PS     = 00B2          PSTAT  = 0048          PSW    =%0009
P.AMTP= 0001           P.BCTC= 0040           P.CMDP= 0020           P.INTE= 0080
P.LCS  = 0040          P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004
P.RPST= 0002           P.RPOE= 0020           P.RP1E= 0010           P.RP2E= 0020
P.RP3E= 0010           P.SING= 0080           P.STAT= 0002           P.STPE= 0080
P.TACH= 0008           P.TUPR= 0010           P.WCSP= 0004           P.WDS  = 0040
P.WFLP= 0001           P.WPEN= 0010           P.WPOE= 0008           P.WP1E= 0004
P.WP2E= 0008           P.WP3E= 0004           P.5VOK= 0002           QUE    = 281B
QUEM   = 281E          QUIT    4580           QUITW   48E3           QUIT1   45A1
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047           RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D
RCMD   = 0008          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          REND   = 0014
REQST  = 2806          RESCHR= 00D1           REVTST= 0064           REWIND= 0004
RFIFOL= 0008           RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001
RILL   = 0012          RINST  = 000C          RMCTST= 0008           RMK2   = 0013
RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001
RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000
RPF1   = 009D          RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016
RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A
RTIER  = 0030          RTM    = 0005          RUNKI  = 0009          RUPTST= 005E
RWDUNL= 0005           R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040
R.DON  = 0002          R.DRDY= 0010           R.END  = 0010          R.ILL  = 0004
R.JVOK= 0004           R.MK2  = 0008          R.PLOD= 0008           R.PLOO= 0010
R.PLO1= 0020           R.POST= 0020           R.STNM= 0002           R.STOP= 0004
R.STPC= 0001           R.TBJN= 0080           R.TSTD= 0040           R.VOK  = 0080
```

```
R00H  = 0081          R00L  = 0080          R01H  = 0083          R01L  = 0082
R02H  = 0085          R02L  = 0084          R03H  = 0087          R03L  = 0086
R04H  = 0089          R04L  = 0088          R05H  = 008B          R05L  = 008A
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SDDCOM  4739          SDDRA   4654          SDDRAE  465A
SDDRAT  4657          SDDRB   4655          SDDRBE  465B          SDDRBT  4658
SDDRC   4656          SDDRCE  465C          SDDRCT  4659          SELCLR= 0000
SETATA= 00A1          SETDDR  46E8          SID   = 0080          SLFTDD  4727
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          TADR00= 0080          TADR01= 0081          TADR02= 0082
TADR03= 0083          TADR04= 0084          TADR05= 0085          TADR06= 0086
TADR07= 0087          TADR10= 0088          TADR11= 0089          TADR12= 008A
TADR13= 008B          TAMT  = 0044          TASEL = 0080          TCMD  = 0040
TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020
TC.WRT= 0010          TEMP    4F99          TEST1   4300          TEST2   4464
TEST3   47FB          TEST4   48F0          TMF   = 0099          TMRDY = 0040
TRKENA= 00D2          TSET  = 2803          TSTEND= 2818          TSTS  = 0040
TST1AC  433A  G       TST1B   4340          TST1C   437B          TST1CA  4397  G
TST1CB  43A0          TST1CC  43BF  G       TST1CD  43E3  G       TST1D   43EC
TST1DC  43FA  G       TST1E   4403          TST1EC  4411  G       TST1F   441A
TST1FC  4429  G       TST1G   4432          TST1L   4305          TST2B   44A3
TST2C   44DE          TST2CD  44FC  G       TST2D   4505          TST2DC  4513  G
TST2E   451C          TST2EC  452A  G       TST2F   4533          TST2FC  4542  G
TST2G   454B          TST2GO  449F          TST2L   4469          TST2LC  4496  G
TST3A   4852          TST3B   4866          TST3BB  486A          TST3BC  487E  G
TST3C   4887          TST3GO  4814          TST3L   4800          TST3WC  484F
TST4C   496E          TST4GO  4930          TST4L   48F5          TST4LC  4927  G
TST4WC  4968          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PS0J= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002          T1CHK   43C8
T1END   4444          T1W     4359          T1WL    4360          T1WLC   4372  G
T1WL1   4362          T2CHK   44E1          T2END   455D          T2W     44BC
T2WL    44C3          T2WLC   44D5  G       T2WL1   44C5          T2WS    4483
T3END   48AE          T3W     482D          T3WL    4834          T3WLC   4846  G
T3WL1   4836          T4END   4995          T4W     4949          T4WL    4950
T4WLC   4962  G       T4WL1   4952          T4WS    4914          UIBG  = 00A1
VALFC   4F98          VALTB   4F95          VELTST= 005B          WDR.P = 0010
WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000
WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008          W.DIAG= 0002
W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080          W.ERR = 0020
W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004          W.ONES= 0020
W.RESI= 0002          W.REV = 0004          W.ROME= 0010          W.RST = 0001
W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020          X     =%000A
X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001
X.WCLK= 0001          Y     =%000B          .     = 49CA
```

ERRORS DETECTED: 0

```
*MBD1,A78/PTP,MBD1=NLIST,PARAM,MACRO,LIST,MBD1
RUN-TIME: 5 7 0 SECONDS
CORE USED: 10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:06 SEQ 0085
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  WMC2 - WRITE MICROCONTROLLER PART #2
                                .SBTTL  TEST 1 - WMC DATA FORMAT/SKIP COUNT ACCEPTANCE/REJECTION
                                ;ID     WMC2-WRITE MICRO CONTROLLER PART #2
1332  4300                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1333                            ;*WMC FORMAT/SKIP COUNT TESTS
1334  4300                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1335                            ;*THIS TEST CHECKS THE OPERATION OF THE WRITE MICRO CONTROLLER IN
1336                            ;*RESPECT TO ACCEPTANCE OF THE SIX (6) DATA FORMATS.  THIS TEST ALSO
1337                            ;*VERIFIES THAT THE FORMATS ARE TAKEN WITHOUT REGARD FOR THE SKIP COUNT
1338                            ;*SPECIFIED.  THE FOLLOWING TABLE LISTS THE COMBINATIONS USED:
1339                            ;*
1340                            ;*       FORMAT          SKIP CNT            EXP. ERR.
1341                            ;*
1342                            ;*         0               0-17                NONE
1343                            ;*         1               0-17                NONE
1344                            ;*         2               0-17                NONE
1345                            ;*         3               0-17                NONE
1346                            ;*         4               0-17                NONE
1347                            ;*         5               0-17                NONE
1348  4300                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1349                            ;*BGNTST
1350                            ;*  INIT BYTE COUNTER, PAD COUNTER, & ERROR  CODE COUNTER CONTROL
1351                            ;*  SET SYSTEM CLOCK TO 'NORMAL'
1352                            ;*  CLEAR FORMAT COUNT INDEX
1353                            ;*  BGNDO
1354                            ;*  :  CLEAR SKIP COUNT INDEX
1355                            ;*  :  BGNDO
1356                            ;*  :  :  RESTART THE WMC + SYSTEM
1357                            ;*  :  :  WRITE CURRENT FORMAT 'COUNT' TO DATACTL WORD - SELECT PACKING MODE
1358                            ;*  :  :  WRITE REGISTER 02 WITH FORMAT NUMBER
1359                            ;*  :  :  CLEAR BYTE COUNTER, PAD COUNTER, AND ECODE COUNTER
1360                            ;*  :  :  CLEAR RESIDUAL CHARACTER WORD - 'RESCHR'
1361                            ;*  :  :  SET WMCCTL TO 'WRITE' AND 'ENABLE' TO START WMC ROM PROGRAM
1362                            ;*  :  :  SMALL DELAY TO WAIT FOR WMC ROM PGM TO FINISH
1363                            ;*  :  :  IF WMCSTA INDICATES 'DONE'
1364                            ;*  :  :  :  THEN-CONTINUE
1365                            ;*  :  :  :  ELSE-REPORT TIMEOUT ERROR
1366                            ;*  :  :  ENDIF
1367                            ;*  :  :  IF WMCERR HAS NO ERROR BITS SET
1368                            ;*  :  :  :  THEN-CONTINUE
1369                            ;*  :  :  :  ELSE-REPORT ERROR DETECTED AFTER A VALID WRITE COMMAND
1370                            ;*  :  :  ENDIF
1371                            ;*  :  :  DO UNTIL SKIP COUNT INDEX=17(8)
1372                            ;*  :  ENDDO
1373                            ;*  :  DO UNTIL FORMAT COUNT INDEX 6
```

```
                                        ;*  ENDDO
                                        ;*ENDTST
                                        SE
                                        ; *****************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*WMC2 MICRO TEST 01
                                        ;*WMC2 MICRO ERROR 01
                                        ;*WMC2-WMC DATA FORMAT/SKIP COUNT ACCEPTANCE
                                        ;*M8959
                                        ;*WMC NOT DONE AFTER A VALID WRITE 'PE' COMMAND ISSUED
                                        ;*DATA FORMAT = MM
                                        ;*SKIP COUNT  = NN
                                        ;*
                                        ;*WMC2 MICRO TEST 01
                                        ;*WMC2 MICRO ERROR 02
                                        ;*WMC2-WMC DATA FORMAT/SKIP COUNT ACCEPTANCE
                                        ;*M8959
                                        ;*WMC ERRORS BYTE NON ZERO AFTER A VALID WRITE PE COMMAND
                                        ;*ACTUAL = NNNN
                                        ;*DATA FORMAT - MM
                                        ;*SKIP COUNT  = NN
                                        S
                                        ; *****************************************************************************

                                        TEST1:  TESTX   @1
                                                        MVI     A,@1            ;DEFINE THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%WMC2-WMC DATA FORMAT/SKIP COUNT ACCEPTANCE
                                        ;&M8959

                                TST1L:  MVI     A,@70           ;SET BYTE COUNTER FOR 0 LENGTH RECORD
                                        OUT     CNTCTL
                                        MVI     A,@160          ;SELECT PAD COUNTER
                                        OUT     CNTCTL
                                        MVI     A,@260          ;SELECT ERROR CODE COUNTER
                                        OUT     CNTCTL

                                        XRA     A               ;SET THE SYSTEM CLOCK TO 'NORMAL'
                                        OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
                                        MVI     A,@233          ;SET THE DDR TO 'IN'
                                        OUT     DDRCTL

                                        MVI     C,$0            ;CLEAR THE FORMAT COUNT
                                T1LP0:  MVI     B,$0            ;CLEAR THE SKIP COUNT
                                T1LP1:  MVI     A,W.RST         ;LOAD THE WMC RESTART BIT
                                        OUT     WMCCTL          ;RESTART THE WMC
                                        XRA     A               ;CLEAR THE ACCUMULATOR
                                        OUT     WMCCTL          ;CLEAR THE RESTART BIT
                                        MOV     A,C             ;COPY FORMAT COUNT
                                        RLC                     ;SHIFT THE FORMAT OVER
                                        RLC                     ;SHIFT THE FORMAT OVER
                                        RLC                     ;3 BIT POSITIONS
                                        RLC                     ;
```

```
1422  4328  E6  70          7.0          ANI     $70             ;REMOVE UNWANTED BITS
1423  432A  B0              4.0          ORA     B               ;OR IN THE SKIP COUNT
1424  432B  D3  D0         10.0          OUT     DATACTL         ;SET UP THE WRITE MICRO
1425                                                             ;WITH FORMAT/SKIP INFO.
1426
1427  432D                               ROUT    R02H            ;LOAD INTO REG 02 HIGH BYTE
 (1)  432D  D3  85         10.0                  OUT     R02H            ;WRITE AC INTO R02H
 (1)  432F  7F              4.0                  MOV     A,A             ;RETRY LINK
1428  4330  AF              4.0          XRA     A               ;AND CLEAR REG 02 LOW BYTE
1429  4331                               ROUT    R02L
 (1)  4331  D3  84         10.0                  OUT     R02L            ;WRITE AC INTO R02L
 (1)  4333  7F              4.0                  MOV     A,A             ;RETRY LINK
1430  4334                               CLRECT                  ;CLEAR THE ERROR COUNTER
 (1)  4334  AF              4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4335  D3  D6         10.0                  OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4337  D3  D6         10.0                  OUT     ERRCNT          ;CLEAR BITS 15-8
1431  4339                               CLRBCT                  ;CLEAR THE BYTE COUNTER
 (1)  4339  AF              4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  433A  D3  D4         10.0                  OUT     BYTCNT          ;CLEAR BITS 7-0
 (1)  433C  D3  D4         10.0                  OUT     BYTCNT          ;CLEAR BITS 15-8
1432  433E                               CLRPCT                  ;CLEAR THE PAD COUNTER
 (1)  433E  AF              4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  433F  D3  D5         10.0                  OUT     PADCNT          ;CLEAR BITS 7-0
 (1)  4341  D3  D5         10.0                  OUT     PADCNT          ;CLEAR BITS 15-8
1433  4343  AF              4.0          XRA     A               ;CLEAR THE ACCUMULATOR
1434  4344  D3  D1         10.0          OUT     RESCHR          ;CLEAR THE RESIDUAL CHARACTER
1435
1436  4346  3E  88          7.0          MVI     A,W.ENAB!W.WRITE  ;
1437  4348  D3  D3         10.0          OUT     WMCCTL          ;ENABLE WMC-WRITE & PE
1438                                                             ;
1439  434A  3E  32          7.0          MVI     A,50            ;SET THE TIMEOUT VALUE
1440  434C  3D              4.0  T1W:    DCR     A               ;DECREMENT THE COUNT
1441  434D  C2  4C  43      10.0          JNZ     T1W             ;CONTINUE UNTIL END OF COUNT
1442
1443  4350  DB  D0         10.0          IN      WMCSTA          ;GET THE WMC STATUS
1444  4352  E6  40          7.0          ANI     W.DONN          ;WMC NOT DONE BIT SET?
1445  4354  CA  5C  43      10.0          JZ      T1CO2           ;NO - CONTINUE
1446  4357                               ERR     T1LP1,T1CO2,@2
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4357  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  435A  01                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  435B  02                                  .BYTE   @2
 (1)  435C  CD  15  28      18.0          T1CO2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  435F  DA  1C  43      10.0                  JC      T1LP1           ;LOOP ADDRESS IF LOOP SPECIFIED
1447                                ;>WMC NOT DONE AFTER A VALID WRITE 'PE' COMMAND ISSUED
1448  4362  DB  DA         10.0          IN      WMCERR          ;GET THE WMC ERRORS BYTE
1449  4364  E6  F8          7.0          ANI     $F8             ;TEST FOR ERROR
1450  4366  CA  71  43      10.0          JZ      T1CN8           ;NONE-CONTINUE
1451  4369                               ROUT    ADATA           ;STORE THE ACTUAL WMC ERROR BYTE
 (1)  4369  D3  94         10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  436B  7F              4.0                  MOV     A,A             ;RETRY LINK
```

```
1453  436C                                        ERRA    T1LP1,T1CN8,@2
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  436C  CD  0F  28        18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0002                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  436F  02                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4370  02                                           .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  4371  CD  15  28        18.0               T1CN8:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4374  DA  1C  43        10.0                       JC      T1LP1           ;LOOP ADDRESS IF LOOP SPECIFIED
1454                                        ;>WMC ERRORS BYTE NON ZERO AFTER A VALID WRITE PE COMMAND
1455  4377  04                    4.0               INR     B               ;UPDATE THE SKIP COUNT
1456  4378  78                    4.0               MOV     A,B             ;
1457  4379  FE  10                7.0               CPI     @20             ;DONE?
1458  437B  C2  1C  43        10.0                   JNZ     T1LP1           ;NO - RUN NEXT SKIP COUNT TEST
1459  437E  0C                    4.0               INR     C               ;INCREMENT THE FORMAT COUNT
1460  437F  79                    4.0               MOV     A,C             ;
1461  4380  FE  06                7.0               CPI     $6              ;DONE?
1462  4382  C2  1A  43        10.0                   JNZ     T1LP0           ;NO-CONTINUE
1463  4385                                          ENDTST  TST1L
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4385                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4385  CD  06  28        18.0                       CALL    REQST
 (2)  4388  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4389  00  00                                       .WORD                   ;SYSTEM '=' COUNT
 (2)  438B  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  438D  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  438E  07                                           .BYTE   7               ;REQUEST CODE
 (1)  438F  3A  9A  4F        13.0               LDA     ITERA           ;GET ITERATION COUNT
 (1)  4392  3D                    4.0               DCR     A               ;DOWNCOUNT
 (1)  4393  32  9A  4F        13.0               STA     ITERA           ;SAVE COUNT
 (1)  4396  F2  05  43        10.0               JP      TST1L           ;DO TEST UNTIL TILL = 0
```

```
                        .SBTIL  TEST 2 - WMC FORMAT/SKIP COUNT CHECKS
                        ST
                        ;*********************** ************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*WMC FORMAT/SKIP COUNT ERROR CHECKING
                        SD
                        ;*************************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST CHECKS THE OPERATION OF THE WRITE MICRO CONTROLLER IN
                        ;*RESPECT TO ACCEPTANCE AND REJECTION OF VARIOUS DATA FORMAT, SKIP
                        ;*COUNT OPERATIONS IN THE READ MODE.  THE RESULTS EXPECTED ARE
                        ;*LISTED BELOW:
                        ;*
                        ;*      FORMAT                  SKIP CNT                EXP. ERR.
                        ;*       0                        0-1                    NONE
                        ;*       0                        2-17                   -2
                        ;*
                        ;*       1                        0-1                    NONE
                        ;*       1                        2-17                   -2
                        ;*
                        ;*       2                        0-4                    NONE
                        ;*       2                        5-17                   -1
                        ;*
                        ;*       3                        0-4                    NONE
                        ;*       3                        5-17                   -1
                        ;*
                        ;*       4                        0-10                   NONE
                        ;*       4                        11-17                  -1
                        ;*
                        ;*       5                        0-17                   NONE
                        ;*
                        ;*       6                        0-10                   NONE
                        ;*       6                        11-17                  -1
                        SP
                        ;*************************************************************************
                        ;*PROCEDURE
                        ;*---------
                        ;*BGNTST
                        ;*  SET PACKING MODE NUMBER TO 0 (11-NORMAL)
                        ;*  SET SKIP COUNT ERROR LIMIT TO 2
                        ;*  SET THE ECODE EXPECTED TO -2
                        ;*  CALL SKIPSUB
                        ;*
                        ;*  SET PACKING MODE NUMBER TO 1 (15-NORMAL)
                        ;*  SET SKIP COUNT ERROR LIMIT TO 2
                        ;*  SET THE ECODE EXPECTED TO -2
                        ;*  CALL SKIPSUB
                        ;*
                        ;*  SET PACKING MODE NUMBER TO 2 (10-COMPAT)
                        ;*  SET SKIP COUNT ERROR LIMIT TO 4
                        ;*  SET THE ECODE EXPECTED TO -1
                        ;*  CALL SKIPSUB
```

```
1510                                ;*
1511                                ;*  SET THE PACKING MODE NUMBER TO 3 (10-DUMP)
1512                                ;*  SET SKIP COUNT ERROR LIMIT TO 5
1513                                ;*  SET THE ECODE EXPECTED TO -1
1514                                ;*  CALL SKIPSUB
1515                                ;*
1516                                ;*  SET PACKING MODE NUMBER TO 4 (10-HIGH DENSITY DUMP)
1517                                ;*  SET SKIP COUNT ERROR LIMIT TO 11(8)
1518                                ;*  SET THE ECODE EXPECTED TO -1
1519                                ;*  CALL SKIPSUB
1520                                ;*
1521                                ;*  SET PACKING MODE NUMBER TO 5 (IMAGE)
1522                                ;*  SET SKIP COUNT ERROR LIMIT TO 0 - SKIP COUNTS CAUSE NO ERROR'
1523                                ;*  CLEAR ECODE EXPECTED VALUE
1524                                ;*  CALL SKIPSUB
1525                                ;*
1526                                ;*  SET PACKING MODE NUMBER TO 6 (10-HIGH DENSITY COMPAT)
1527                                ;*  SET SKIP COUNT ERROR LIMIT TO 11(8)
1528                                ;*  SET ECODE EXPECTED TO -1
1529                                ;*  CALL SKIPSUB
1530                                ;*ENDTST
1531  4399                          SE
 (1)                                ;*****************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1532                                ;*WMC2 MICRO ERROR 02
1533                                ;*WMC2 MICRO ERROR 03
1534                                ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
1535                                ;*M8959
1536                                ;*WMC ERRORS BYTE NON ZERO AFTER A VALID READ COMMAND
1537                                ;*ACTUAL = NNNN
1538                                ;*DATA FORMAT - MM
1539                                ;*SKIP COUNT  - NN
1540                                ;*
1541                                ;*WMC2 MICRO ERROR 02
1542                                ;*WMC2 MICRO ERROR 04
1543                                ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
1544                                ;*M8959
1545                                ;*WMC - ERRORS BYTE INCORRECT AFTER INVALID READ COMMAND
1546                                ;*ACTUAL = NNNN
1547                                ;*EXPECTED = NNNN
1548                                ;*DATA FORMAT = MM
1549                                ;*SKIP COUNT  = NN
1550                                ;*
1551                                ;*WMC2 MICRO ERROR 02
1552                                ;*WMC2 MICRO ERROR 05
1553                                ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
1554                                ;*M8959
1555                                ;*WMC ECODE COUNTER LOW BYTE NOT CORRECT AFTER INVALID READ COMMAND
1556                                ;*ACTUAL - NNNN
1557                                ;*EXPECTED   NNNN
1558                                ;*DATA FORMAT   MM
1559                                ;*SKIP COUNT    NN
1560                                ;*
```

```
1561                                    ;*WMC2 MICRO ERROR 02
1562                                    ;*WMC2 MICRO ERROR 06
1563                                    ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
1564                                    ;*M8959
1565                                    ;*WMC ECODE COUNTER HIGH BYTE NOT CORRECT AFTER INVALID READ COMMAND
1566                                    ;*ACTUAL = NNNN
1567                                    ;*EXPECTED = NNNN
1568                                    ;*DATA FORMAT = MM
1569                                    ;*SKIP COUNT  = NN
1570  4399                              $
 (1)                                    ; *****************************************************************************
1571
1572  4399                              TEST2:  TESTX   @2
 (1)  4399  3E  02          7.0                         MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4398  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
1573                                    ;%WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
1574                                    ;@M8959
1575
1576  439E  3E  00          7.0  TST2L:  MVI     A,@0            ;LOAD PDP11 NORMAL FORMAT CODE
1577  43A0  32  1D  44     13.0          STA     FMTCOD          ;STORE FORMAT CODE
1578  43A3  3E  02          7.0          MVI     A,@2            ;LOAD THE FIRST FAIL SKIP COUNT
1579  43A5  32  1C  44     13.0          STA     SKIPFF          ;STORE SKIP COUNT
1580  43A8  21  FE  FF     10.0          LXI     H,@-2           ;LOAD THE SKIP ERROR CODE
1581  43AB  22  1E  44     16.0          SHLD    ERRCOD          ;STORE ERROR CODE
1582  43AE  CD  20  44     18.0          CALL    SKIPSUB         ;
1583                                                            ;
1584  43B1  3E  01          7.0          MVI     A,@1            ;LOAD PDP15 NORMAL FORMAT CODE
1585  43B3  32  1D  44     13.0          STA     FMTCOD          ;STORE FORMAT CODE
1586                                  ;FIRST FAIL SKIP COUNT IS 2, AND IS STILL SET UP FROM ABOVE
1587                                  ;ERROR CODE IS -2, AND IS STILL SET UP FROM ABOVE
1588  43B6  CD  20  44     18.0          CALL    SKIPSUB         ;
1589                                                            ;
1590  43B9  3E  02          7.0          MVI     A,@2            ;LOAD PDP10 COMPAT. FORMAT CODE
1591  43BB  32  1D  44     13.0          STA     FMTCOD          ;STORE FORMAT CODE
1592  43BE  3E  04          7.0          MVI     A,@4            ;LOAD THE FIRST FAIL SKIP COUNT
1593  43C0  32  1C  44     13.0          STA     SKIPFF          ;STORE SKIP COUNT
1594  43C3  21  FF  FF     10.0          LXI     H,@-1           ;LOAD THE SKIP FAIL ERROR CODE
1595  43C6  22  1E  44     16.0          SHLD    ERRCOD          ;STORE THE ERROR CODE
1596  43C9  CD  20  44     18.0          CALL    SKIPSUB         ;
1597                                                            ;
1598  43CC  3E  03          7.0          MVI     A,@3            ;LOAD PDP10 DUMP FORMAT CODE
1599  43CE  32  1D  44     13.0          STA     FMTCOD          ;STORE THE FORMAT CODE
1600  43D1  3E  05          7.0          MVI     A,@5            ;FIRST FAIL OF 5
1601  43D3  32  1C  44     13.0          STA     SKIPFF          ;FOR 10 DUMP MODE - SKIP COUNT
1602
1603                                  ;ERROR CODE IS -1, AND IS STILL SET UP FROM ABOVE
1604
1605  43D6  CD  20  44     18.0          CALL    SKIPSUB         ;
1606                                                            ;
1607  43D9  3E  04          7.0          MVI     A,@4            ;LOAD PDP10 HIGH DEN. FORMAT CODE
1608  43DB  32  1D  44     13.0          STA     FMTCOD          ;STORE THE FORMAT CODE
1609  43DE  3E  09          7.0          MVI     A,@11           ;LOAD THE FIRST FAIL SKIP COUNT
1610  43E0  32  1C  44     13.0          STA     SKIPFF          ;STORE THE SKIP COUNT
1611
```

```
1612                                        ;ERROR CODE IS -1, AND IS STILL SET UP FROM ABOVE
1613
1614  43E3  CD  20  44      18.0            CALL    SKIPSUB         ;
1615                                        ;
1616  43E6  3E  05           7.0            MVI     A,@5            ;LOAD IMAGE FORMAT CODE
1617  43E8  32  1D  44      13.0            STA     FMTCOD          ;STORE THE FORMAT CODE
1618  43EB  AF               4.0            XRA     A               ;INDICATE NO FAILING SKIP COUNTS
1619  43EC  32  1C  44      13.0            STA     SKIPFF          ;STORE THE SKIP COUNT
1620  43EF  CD  20  44      18.0            CALL    SKIPSUB
1621  43F2  3E  06           7.0            MVI     A,@6
1622  43F4  32  1D  44      13.0            STA     FMTCOD
1623  43F7  3E  09           7.0            MVI     A,@11
1624  43F9  32  1C  44      13.0            STA     SKIPFF
1625  43FC  21  FF  FF      10.0            LXI     H,@-1
1626  43FF  22  1E  44      16.0            SHLD    ERRCOD
1627  4402  CD  20  44      18.0            CALL    SKIPSUB
1628
1629  4405                                  ENDTST  TST2L
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4405                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4405  CD  06  28      18.0                            CALL    REQST
 (2)  4408  00                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  4409  C0  00                                          .WORD                   ;SYSTEM '0' COUNT
 (2)  440B  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  440D  0C                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  440E  07                                              .BYTE   7               ;REQUEST CODE
 (1)  440F  3A  9A  4F      13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  4412  3D               4.0            DCR     A               ;DOWNCOUNT
 (1)  4413  32  9A  4F      13.0            STA     ITERA           ;SAVE COUNT
 (1)  4416  F2  9E  43      10.0            JP      TST2L           ;DO TEST UNTIL TILL = 0
1630  4419                                  S
 (1)                                  ;*****************************************************************************
1631  4419  C3  18  28      10.0            JMP     TSTEND
1632  441C                                  S
 (1)                                  ;*****************************************************************************
1633  441C  00                      SKIPFF: .BYTE   0               ;FIRST FAIL SKIP COUNT
1634  441D  00                      FMTCOD: .BYTE   0               ;DATA FORMAT MODE UNDER TEST
1635  441E  00  00                  ERRCOD: .WORD   0               ;ERROR CODE EXPECTED
```

```
                        .SBTTL  SKIPSUB - SUBROUTINE FOR TEST 2
                        SSUB
                        ;*****************************************************************************
                        ;*SUBROUTINE TITLE
                        ;*---------- -----
                        ;*SKIP TEST SUBROUTINE
                        SD
                        ;*****************************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS SUBROUTINE PERFORMS ALL THE INPUT/OUT AND ERROR REPORTING FOR
                        ;*TEST 2.  THE SUBROUTINE LOADS THE DATA FORMAT SPECIFIED WITH SKIP
                        ;*COUNTS FROM 0-17(8) AND CHECKS THE RESPONSE OF THE WRITE MICRO
                        ;*CONTROLLER TO THE DATA FORMAT/SKIP COUNT BYTE, ANY DEVIATION FROM
                        ;*THE EXPECTED RESULTS CAUSES AN ERROR MESSAGE TO BE OUTPUT.
                        ;*
                        ;*      INPUT:
                        ;*
                        ;*      FMTCOD  THIS DATA BYTE CONTAINS THE NUMBER OF THE DATA FORMAT
                        ;*              TO BE TESTED AS FOLLOWS:
                        ;*                      0 = 11 NORMAL
                        ;*                      1 = 15 NORMAL
                        ;*                      2 = 10 COMPATABILITY
                        ;*                      3 = 10 CORE DUMP
                        ;*                      4 = 10 HIGH DENSITY
                        ;*                      5 - IMAGE
                        ;*
                        ;*      SKIPFF  THIS DATA BYTE CONTAINS THE FIRST FAIL SKIP COUNT, WHICH
                        ;*              IS THE FIRST ILLEGAL SKIP COUNT FOR THE DATA FORMAT.
                        ;*
                        ;*      ERRCOD  THIS DATA WORD (16 BITS) CONTAINS THE NUMBER THAT SHOULD
                        ;*              BE PRESENT IN THE WRITE MICRO CONTROLLER ECODE COUNTER FOR
                        ;*              THE ILLEGAL SKIP COUNTS FOR THE DATA FORMAT.
                        ;*
                        ;*      OUTPUT:
                        ;*
                        ;*              THIS SUBROUTINE HAS NO OUTPUT TO RETURN TO THE CALLING
                        ;*              MODULE.  HOWEVER, AS A RESULT OF ERROR CONDITIONS DETECTED
                        ;*              WITHIN THIS MODULE, ERROR MESSAGES HANDSHAKES WITH THE
                        ;*              HOST CAN TAKE PLACE.
                        S
                        ;*****************************************************************************
                        SP
                        ;*****************************************************************************
                        ;*PROCEDURE
                        ;*---------
                        ;*BGNSUB
                        ;*  INIT ALL COUNTER CONTROL-BYTE, PAD, + ECODE
                        ;*  SET SKIP COUNT INDEX TO 0
                        ;*  BGNDO
                        ;*  :  RESTART WMC
                        ;*  :  LOAD WMC PACKING MODE NUMBER INTO DATACTL
                        ;*  :  CLEAR ALL COUNTERS-BYTE, PAD, + ECODE
                        ;*  :  CLEAR RESCHR WORD
```

```
                                        ;*   :  ENABLE WMC SO ROM PGM WILL START CHECKING PARAMETERS
                                        ;*   :  SMALL DELAY FOR WMC ROM PGM TO CHECK PARAMETERS
                                        ;*   :  CLEAR WMC ENABLE BIT SO WMC 'DONE' WILL SET
                                        ;*   :  IF FAIL ON SKIP COUNT EXPECTED
                                        ;*   :  :  THEN-CHECK FAIL COUNT EXPECTED
                                        ;*   :  :  IF CURRENT SKIP COUNT > OR = FAIL COUNT EXPECTED
                                        ;*   :  :  :  THEN-CHECK FOR EXPECTED ERROR CONDIDION
                                        ;*   :  :  :  IF ERROR AS EXPECTED
                                        ;*   :  :  :  :  THEN-CONTINUE
                                        ;*   :  :  :  :  ELSE-REPORT INCORRECT ERROR ON INVALID READ COMMAND
                                        ;*   :  :  :  ENDIF
                                        ;*   :  :  :  ELSE-CONTINUE
                                        ;*   :  :  ENDIF
                                        ;*   :  :  ELSE-CHECK FOR NO ERROR
                                        ;*   :  :  IF WMC DETECTED ERROR
                                        ;*   :  :  :  THEN-REPORT ERROR AFTER A VALID READ COMMAND
                                        ;*   :  :  :  ELSE-CONTINUE
                                        ;*   :  :  ENDIF
                                        ;*   :  ENDIF
                                        ;*   :  DO UNTIL SKIP COUNT > 17(8)
                                        ;*  ENDDO
                                        ;*ENDSUB
1703  4420                              S
 (1)                                    ; ***************************************************************************************
1704
1705  4420                              SKIPSUB: INICNT                 ;INITIALIZE THE COUNTERS
 (1)  4420  3E  30            7.0                        MVI     A,@060          ;INIT THE
 (1)  4422  D3  D7           10.0                        OUT     CNTCTL          ;BYTE COUNTER
 (1)  4424  3E  70            7.0                        MVI     A,@160          ;INIT THE
 (1)  4426  D3  D7           10.0                        OUT     CNTCTL          ;PAD COUNTER
 (1)  4428  3E  B0            7.0                        MVI     A,@260          ;INIT THE
 (1)  442A  D3  D7           10.0                        OUT     CNTCTL          ;ERROR COUNTER
1706  442C  06  00             7.0          SKLP0:        MVI     B,0             ;CLEAR THE SKIP COUNT OFFSET POINTER
1707  442E  3E  01             7.0                        MVI     A,W.RST         ;LOAD THE WMC RESTART COMMAND BIT
1708  4430  D3  D3            10.0                        OUT     WMCCTL          ;RESTART THE WMC
1709  4432  AF                 4.0                        XRA     A               ;CLEAR THE ACCUMULATOR
1710  4433  D3  D3            10.0                        OUT     WMCCTL
1711  4435  3A  1D  44         13.0                        LDA     FMTCOD          ;GET THE FORMAT CODE
1712  4438  07                 4.0                        RLC                     ;SHIFT THE ACC. LEFT
1713  4439  07                 4.0                        RLC                     ;3 BIT POSITIONS
1714  443A  07                 4.0                        RLC                     ;3 BIT POSITIONS
1715  443B  07                 4.0                        RLC
1716  443C  E6  70             7.0                        ANI     $70             ;REMOVE UNWANTED BITS
1717  443E  B0                 4.0                        ORA     B               ;OR IN SKIP COUNT
1718  443F  D3  D0            10.0                        OUT     DATACTL         ;SET UP THE WRITE MICRO
1719  4441                               ROUT    R02L                            ;WITH FORMAT/SKIP INFO.
 (1)  4441  D3  84            10.0                                OUT     R02L            ;WRITE AC INTO R02L
 (1)  4443  7F                4.0                                 MOV     A,A             ;RETRY LINK
1720  4444                               CLRECT                          ;CLEAR THE ECODE COUNT
 (1)  4444  AF                4.0                                 XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4445  D3  D6           10.0                                OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4447  D3  D6           10.0                                OUT     ERRCNT          ;CLEAR BITS 15-8
1721  4449                               CLRBCT                          ;CLEAR THE BYTE COUNT
 (1)  4449  AF                4.0                                 XRA     A               ;CLEAR THE ACCUMULATOR
```

```
                                        OUT     BYTCNT          ;CLEAR BITS 7-0
                                        OUT     BYTCNT          ;CLEAR BITS 15-8
                                CLRPCT          ;CLEAR THE PAD COUNT
                                        XRA     A               ;CLEAR THE ACCUMULATOR
                                        OUT     PADCNT          ;CLEAR BITS 7-0
                                        OUT     PADCNT          ;CLEAR BITS 15-8
                                XRA     A                       ;CLEAR THE ACCUMULATOR
                                OUT     RESCHR                  ;CLEAR THE RESIDUAL CHARACTER
                                ;
                                MVI     A,W.ENAB                ;
                                OUT     WMCCTL                  ;ENABLE WMC - READ PE
                                ;
                                MVI     A,30                    ;LOAD THE TIMEOUT COUNT
                        T2W:    DCR     A                       ;DECREMENT THE TIMEOUT COUNT
                                JNZ     T2W                     ;CONTINUE UNTIL TIMEOUT
                                XRA     A
                                OUT     WMCCTL                  ;NOW DISABLE WMC SO 'DONE' WILL SET
                                LDA     SKIPFF                  ;GET THE FIRST FAIL SKIP FLAG
                                ANA     A                       ;SET CONDITION CODE
                                JZ      SKIP0                   ;NO ERROR EXPECTED-GO CHECK
                                CMP     B                       ;COMPARE SKIP COUNT & FF SKIP COUNT
                                JC      SKIP1                   ;SKIP COUNT > FF SKIP COUNT
                                JZ      SKIP1                   ;SKIP COUNT IS EQUAL TO THE FIRST FAIL SKIP COUNT
                        SKIP0:
                                IN      WMCERR                  ;GET THE WMC ERRORS BYTE
                                ANI     $70                     ;TEST FOR ERROR
                                JZ      SKCN3                   ;NONE-CONTINUE
                                ROUT    ADATA                   ;STORE THE ACTUAL WMC ERROR BYTE
                                        OUT     ADATA                   ;WRITE AC INTO ADATA
                                        MOV     A,A                     ;RETRY LINK
                                ERRA    SKLP0,SKCN3,@2
                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
                                        CALL    ERLPA                   ;PROCESS ERROR - DO 2.3
                                        MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
                                        .BYTE   MSGN                    ;MESSAGE NUMBER ID
                                        .BYTE   @2                      ;PRINT ROUTINE NUMBER
                                SKCN3:: CALL    CKLOP                   ;CHECK LOOP FUNCTION - DO 2.2
                                        JC      SKLP0                   ;LOOP ADDRESS IF LOOP SPECIFIED
                        ;>WMC ERRORS BYTE NON ZERO AFTER A VALID READ COMMAND
                                JMP     SKINC
                        SKIP1:
                                IN      WMCERR                  ;INPUT THE WMC ERRORS BYTE
                                ANI     $70                     ;REMOVE UNWANTED BITS
                                CPI     W.ROME!W.ERR            ;WMC ROM PARITY ERROR AND ERROR?
                                JZ      SKCN7                   ;YES-CONTINUE
                                ROUT    ADATA                   ;STORE ACTUAL ERROR BYTE
                                        OUT     ADATA                   ;WRITE AC INTO ADATA
                                        MOV     A,A                     ;RETRY LINK
                                MVI     A,W.ROME!W.ERR          ;LOAD EXPECTED ERROR BYTE
                                ROUT    EDATA                   ;STORE IN CAS
                                        OUT     EDATA                   ;WRITE AC INTO EDATA
                                        MOV     A,A                     ;RETRY LINK
```

```
1757  449A                                          ERRB    SKLP0,SKCN7,@2
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  449A  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  449D  04                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  449E  02                                              .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  449F  CD  15  28        18.0                  SKCN7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44A2  DA  2E  44        10.0                          JC      SKLP0           ;LOOP ADDRESS IF LOOP SPECIFIED
1758                                        ;>WMC - ERRORS BYTE INCORRECT AFTER INVALID READ COMMAND
1759  44A5  2A  1E  44        16.0          LHLD    ERRCOD          ;EXPECTED ECODE COUNT=ACTUAL ECODE COUNT
1760  44A8  7D                  4.0          MOV     A,L             ;COPY L TO A
1761  44A9                                   ROUT    EDATA           ;STORE EXPECTED ECODE COUNT
 (1)  44A9  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44AB  7F                  4.0                  MOV     A,A             ;RETRY LINK
1762  44AC  DB  D6            10.0          IN      ERRCNT          ;INPUT ECODE COUNT BITS 7:0
1763  44AE                                   ROUT    ADATA           ;STORE ACTUAL ECODE COUNT
 (1)  44AE  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44B0  7F                  4.0                  MOV     A,A             ;RETRY LINK
1764  44B1  BD                  4.0          CMP     L               ;COMPARE ACTUAL AND EXPECTED
1765  44B2  CA  BA  44          10.0          JZ      SKCN5           ;CONTINUE-IF EQUAL
1766  44B5                                   ERRB    SKLP0,SKCN5,@2
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44B5  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                                            MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B8  05                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B9  02                                              .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  44BA  CD  15  28        18.0                  SKCN5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44BD  DA  2E  44        10.0                          JC      SKLP0           ;LOOP ADDRESS IF LOOP SPECIFIED
1767                                        ;>WMC ECODE COUNTER LOW BYTE NOT CORRECT AFTER INVALID READ COMMAND
1768  44C0  7C                  4.0          MOV     A,H             ;COPY H TO A
1769  44C1                                   ROUT    EDATA           ;STORE EXPECTED ECODE COUNT
 (1)  44C1  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44C3  7F                  4.0                  MOV     A,A             ;RETRY LINK
1770  44C4  DB  D6            10.0          IN      ERRCNT          ;INPUT ECODE COUNT BITS 15:8
1771  44C6                                   ROUT    ADATA           ;STORE ACTUAL ECODE COUNT
 (1)  44C6  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44C8  7F                  4.0                  MOV     A,A             ;RETRY LINK
1772  44C9  BC                  4.0          CMP     H               ;COMPARE ACTUAL WITH EXPECTED
1773  44CA  CA  D2  44          10.0          JZ      SKCN6           ;CONTINUE IF EQUAL
1774  44CD                                   ERRB    SKLP0,SKCN6,@2
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44CD  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D0  06                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D1  02                                              .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  44D2  CD  15  28        18.0                  SKCN6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D5  DA  2E  44        10.0                          JC      SKLP0           ;LOOP ADDRESS IF LOOP SPECIFIED
1775                                        ;>WMC ECODE COUNTER HIGH BYTE NOT CORRECT AFTER INVALID READ COMMAND
```

```
1777                                        ;HERE TO UPDATE THE SKIP COUNT FOR THE NEXT TEST
1778
1779  44D8  04                      4.0     SKINC:  INR     B               ;UPDATE THE SKIP COUNT
1780  44D9  78                      4.0             MOV     A,B             ;
1781  44DA  FE  10                  7.0             CPI     @20             ;DONE?
1782  44DC  C2  2E  44              10.0            JNZ     SKLPC           ;NO - CONTINUE
1783  44DF  C9                      10.0            RET                     ;YES - RETURN
1784
1785        0000                                    .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE    4F9B          ATTCD    4F97         AXNUM    4F91
B      =%0000          BADST  = 0090          BIT0   = 0001         BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008         BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080         BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE    4F9C          BYTCNT= 00D4          BYTEH    4F24
BYTEL    4F23          C      =%0001          CASCT    4F21         CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088         CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087         CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLKCTL= 00F0           CLOCK    4F26          CMC0H  = 0099         CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D         CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097         CMINL  = 0096
CNTCTL= 00D7           CRCWRD= 0018           CSAVE    4F9D         CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084         CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080         CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008         C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001         C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC         C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002
D      =%0002          DATACT= 00D0           DBUS     4F28         DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DIAFLG   4F22          DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 0008          DONE1  = 0045          DONINT= 0010          DSAVE    4F9E
DSE    = 0006          D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          E      =%0003          ECCBAD= 0042          ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095
EOTCLR= 0003           ERFLG    4F93          ERLP   = 2809         ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM    4F90         ERRCNT= 00D6
ERRCOD   441E          ESAVE    4F9F          E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002           E.UNC  = 0004         FIFORD= 006A
FMTCOD   441D          FORMAT   4F25          FWDTST= 0061          GCRID  = 0089
GCRSET= 0002           GOODTM= 0092           H      =%0004         HLSAVE   4FA0
IE     = 0008          INTSTA= 00E0           ITERA    4F9A         I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020         I7.5   = 0040
KCALL  = 005F          KCLR   = 007B          KDEP   = 003F         KENAB  = 0078
KEXAM  = 003E          KEYBRD= 00C8           KEY1   = 0078         KEY10  = 006D
KEY11  = 006E          KEY12  = 006F          KEY13  = 005C         KEY14  = 005D
KEY15  = 005E          KEY16  = 005F          KEY17  = 003C         KEY18  = 003D
KEY19  = 003E          KEY2   = 0079          KEY20  = 003F         KEY3   = 007A
KEY4   = 007B          KEY5   = 0074          KEY6   = 0075         KEY7   = 0076
KEY8   = 0077          KEY9   = 006C          KINTA  = 006F         KLDAD  = 003D
KN0    = 003C          KN1    = 005C          KN2    = 005D         KN3    = 005E
KN4    = 006C          KN5    = 006D          KN6    = 006E         KN7    = 0074
KN8    = 0075          KN9    = 0076          KU2    = 0079         KU3    = 007A
KU8    = 0077          L      =%0005          LBLANK= 000F          LCE    = 000B
```

```
LCH    = 000C          LCL    = 000D          LCP    = 000E          LC0    = 0000
LC1    = 0001          LC2    = 0002          LC3    = 0003          LC4    = 0004
LC5    = 0005          LC6    = 0006          LC7    = 0007          LC8    = 0008
LC9    = 0009          LDLEDA= 00CA           LDLEDB= 00CB          LDLEDC= 00CC
LDLEDD= 00CD           LDLEDE= 00CE           LDLEDF= 00CF          LKDIAG= 2800
LKKBD  = 004C          LKKEY  = 0049          LKLWMG= 0058          LKLWMP= 0055
LKLWPG= 0052           LKLWPP= 004F           LKMOD7= 0046          LKOPR  = 0046
LPFLG   4F94           LPNUM   4F92           M      =%0006         MBSEL  = 00E0
MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF          MINUS  = 000A
MM     = 8000          MSE    = 0008          MSGN   = 0006         MTACLR= 0000
MT.ARA= 0020           MT.CPE= 0080           MT.DSE= 0001          MT.FWD= 0040
MT.INH= 0008           MT.LWR= 0004           MT.MOT= 0002          MT.NWT= 0080
MT.PEC= 0040           MT.PSB= 0004           MT.PS0= 0001          MT.PS1= 0002
MT.REV= 0020           MT.WRT= 0010           MT.Z   - 0008         M.ATA  = 0080
M.CAPE= 0020           M.CONT= 0080           M.DEM  = 0020         M.EBL  = 0010
M.EXC  = 0008          M.FAIL= 0008           M.ILR  - 0010         M.INIT= 0010
M.OCC  = 0020          M.ONLI= 0001           M.PE   = 0040         M.PORT= 0080
M.RDEN= 0002           M.RDPE= 0008           M.RUN  = 0004         M.SCLK= 0001
M.TRA  = 0040          M.UNIT= 0007           M.WCLK= 0040          M.WCLN= 0080
M.WREN= C080           M5.5   = 0001          M6.5   = 0002         M7.5   = 0004
NOTCAP= 0088           OKAY   = 00FF          OPRRAM= 4300          OPSTRT= 0058
OPVER  = 0040          PADCNT- 00D5           PADCRC= 0080          PDIAG  = 0048
PEID   - 008A          PENAB  = 004C          PESET  = 0001         PL     = 00B1
PRDD   = 004C          PRENF  = 009C          PS     = 00B2         PSTAT  - 0048
PSW    =%0009          P.AMTP= 0001           P.BCTC= 0040          P.CMDP= 0020
P.INTE= 0080           P.LCS  - 0040          P.LWR  = 0020         P.RDP  = 0002
P.RPEN= 0004           P.RPST= 0002           P.RPOE= 0020          P.RP1E= 0010
P.PP2E= 0020           P.RP3E- 0010           P.SING= 0080          P.STAT= 0002
P.STPE= 0080           P.TACH= 0008           P.TUPR= 0010          P.WCSP= 0004
P.WDS  = 0040          P.WFLP= 0001           P.WPEN= 0010          P.WP0E= 0008
P.WP1E= 0004           P.WP2E= 0008           P.WP3E= 0004          P.5VOK= 0002
QUE    = 281B          QUEM   = 281E          RAMT   = 0010         RARA   = 0006
RARAI  = 0004          RCHBD0= 0048           RCHBD1= 0047          RCHOK  = 0046
RCHTST= 000C           RCLRT  = 000D          RCMD   = 000B         RCMLP  = 0003
RCONT  = 0080          RDATA  = 0017          RDCLK  = 0010         RDON   = 0011
READG  = 0007          REND   = 0014          REQST  = 2806         RESCHR= 00D1
REVTST= 0064           REWIND  0004           RFIFOL= 0008          RGCLK  = 0002
RGCRI  - 0003          RIBG   = 0001          RILL   = 0012         RINST  = 000C
RMLTST= 0008           RMK2   = 0013          RNOP   = 0000         RPATH  = 0001
RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003         RPCTL  = 0009
RPEI   = 0002          RPFAIL= 0000           RPF1   = 009D         RPF2   = 009E
RPOK   = 0043          RPOSTN= 0016           RPSTA  = 0015         RRCMT  = 000A
RSTAT  = 0002          RTIEB  = 000A          RTIER  = 0030         RTM    = 0005
RUNKI  = 0009          RUPTST= 005E           RWDUNL= 0005          R.AMT  = 0001
R.BOP  = 0008          R.DATA= 0040           R.DON  = 0002         R.DRDY= 0010
R.END   0010           R.ILL  = 0004          R.JVOK= 0004          R.MK2  = 0008
R.PLOD= 0008           R.PLO0= 0010           R.PLO1= 0020          R.POST= 0020
R.STNM= 0002           R.STOP= 0004           R.STPC= 0001          R.TBJN= 0080
R.TSTD= 0040           R.VOK  = 0080          R00H   = 0081         R00L   = 0080
R01H   - 0083          R01L   - 0082          R02H   = 0085         R02L   = 0084
R03H   = 0087          R03L   = 0086          R04H   = 0089         R04L   = 0088
R05H   = 008B          R05L   = 008A          R06H   = 008D         R06L   = 008C
R07H   = 008F          R07L   = 008E          R10H   = 0091         R10L   = 0090
R11H   - 0093          R11L   = 0092          R12H   = 0095         R12L   = 0094
```

WMC2 - WRITE MICROCONTROLLER PART #2 CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:06 PAGE 1-15 SEQ 0101
WMC2.M80 SYMBOL TABLE

```
R13H  = 0097          R13L  = 0096          R14H  = 0099          R14L  = 0098
R15H  = 009B          R15L  = 009A          R16H  = 009D          R16L  = 009C
R17H  = 009F          R17L  = 009E          R7.5  = 0010          SELCLR= 0000
SETATA= 00A1          SID   = 0080          SKCN3   4480  G       SKCN5   44BA  G
SKCN6   44D2  G       SKCN7   449F  G       SKINC   44D8          SKIPFF  441C
SKIPSU  4420          SKIP0   4471          SKIP1   4489          SKLP0   442E
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          TADR00= 0080          TADR01- 0081          TADR02= 0082
TADR03= 0083          TADR04= 0084          TADR05= 0085          TADR06= 0086
TADR07= 0087          TADR10= 0088          TADR11= 0089          TADR12= 008A
TADR13= 0088          TAMT  = 0044          TASEL = 0080          TCMD  = 0040
TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020
TC.WRT= 0010          TEMP    4F99          TEST1   4300          TEST2   4399
TMF   = 0099          TMRDY = 0040          TRKENA= 00D2          TSET  = 2803
TSTEND= 2818          TSTS  = 0040          TST1L   4305          TST2L   439E
TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002          T.FPT = 0001
T.NTHR= 0004          T.ONL = 0020          T.PES = 0008          T.PSBJ= 0020
T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080          T.RDYO= 0040
T.RWD = 0010          T.SCLK= 0002          T1CN8   4371  G       T1CO2   435C  G
T1LP0   431A          T1LP1   431C          T1W     434C          T2W     445C
UIBG  = 00A1          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT- 0004
W.ONES= 0020          W.RESI- 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT  0008          W.XFER- 0020
X     =%000A          X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002
X.ROME- 0001          X.WCLK- 0001          Y     =%000B          .       44E0
```

ERRORS DETECTED: 0

```
*WMC2.A78/PTP,WMC2=NLIST,PARAM,MACRO,LIST,WMC2
RUN-TIME: 3 4 0 SECONDS
CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:06 SEQ 0102
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  WMC3 - BYTE ASSEMBLY TESTS
                                .SBTTL  TEST 1 - WRITE MICRO BYTE ASSY. TESTS FOR ALL FORMAT CODES
                                ;ID     WMC3-WRITE MICRO CONTROLLER PART #3

4300                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*BYTE ASSBMELY TESTS
4300                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SERIES OF TESTS IS DESIGNED TO TEST THE WRITE MICROCONTROLLER BYTE
                                ;*ASSEMBLY LOGIC.  A KNOWN DATA PATTERN IS FED INTO BYTE ASSEMBLY LOGIC
                                ;*FROM THE DIAGNOSTIC DATA REGISTERS - 'DDR' - TO PROVIDE AN 18 BIT WIDE
                                ;*REPLACEMENT FOR MASSBUS DATA.
                                ;*
                                ;*DEPENDING ON THE FORMAT CODE SELECTED, THE BYTE ASSEMBLY LOGIC WILL
                                ;*'BREAK UP' THE INPUT DATA (18 BITS) INTO THE DESIRED BYTE FORMAT TO BE
                                ;*WRITTEN ON TAPE.
                                ;*
                                ;*THIS TEST WILL ISSUE 4 'SCLKS' OF 18 BIT INPUT DATA (FOR EACH FORMAT
                                ;*SELECTED AND EACH DATA TEST PATTERN SELECTED) FOR 72 BITS OF INPUT DATA
                                ;*FROM 'DDR' REGISTERS.  THE SELECTED FORMAT CODES WILL GENERATE THE FOLLOWING
                                ;*AMOUNT OF BAL OUTPUT BYTES FOR EACH 4 SCLKS GENERATED.  TOTAL AMOUNT OF
                                ;*SCLKS/FORMAT=60.
                                ;*
                                ;*FORMAT CODE   ORDER OF BYTES      AMOUNT OF BYTES/4 SCLK CYCLE
                                ;*0-11 NORMAL       B-A                          8
                                ;*1-15 NORMAL       A-B                          8
                                ;*2-10 COMPATABILITY A-B-C-D                      8
                                ;*3-10 DUMP         A-B-C-D-E                   10
                                ;*4-10 HIGH DEN COMP A-B-C-D-E-F-G-H-I            9
                                ;*5-IMAGE            A                           4
                                ;*4-10 HIGH DEN DUMP A-B-C-D-E-F-G-H-I            9
4300                            S
                                ;*****************************************************************************
4300                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  INIT THE TEST
                                ;*  CLEAR FORMAT INDEX NUMBER
                                ;*  BGNDO
                                ;*  :   INIT DATA PATTERN NUMBER FOR PASS
                                ;*  :   RESTART WMC & SET CLOCK CONTROL TO SINGLE STEP
                                ;*  :   INIT WMC BYTE COUNTER + ERROR CODE COUNTER CONTROL
                                ;*  :   CLEAR COUNTERS
                                ;*  :   CLEAR PROGRAM FLAGS AND STORAGE AREAS
                                ;*  :   SET DDR CONTROL TO 'OUT' SO DATA INJECTION POSSIBLE
                                ;*  :   GENERATE DATA TO DDR REGISTERS AND CHECK STORAGE
                                ;*  :   INIT SCLK COUNTER
```

```
1373                    ;*  :  LOAD DATACTL WITH CURRENT FORMAT NUMBER-ALSO SAVE COPY IN REG 02
1374                    ;*  :  CLOCK WMC THROUGH ITS RESTART AREA-WMC ROM PROGRAM
1375                    ;*  :  LOAD BYTE COUNTER
1376                    ;*  :  SET WMCCTL TO 'WRITE' & 'ENABLE'-START WMC ROM PGM
1377                    ;*  :  BGNDO
1378                    ;*  :  :  CLOCK SYSTEM
1379                    ;*  :  :  IF WMCSTA HAS NO 'XFER' BIT SET
1380                    ;*  :  :  :  THEN-UPDATE TIMEOUT COUNTERS
1381                    ;*  :  :  :  :  IF TIMEOUT DETECTED
1382                    ;*  :  :  :  :  :  THEN-REPORT LACK OF ACTIVITY IN SPECIFIED TIME
1383                    ;*  :  :  :  :  :  ELSE-CONTINUE
1384                    ;*  :  :  :  :  ENDIF
1385                    ;*  :  :  :  ELSE-GET WMCSTA WORD + SAVE FOR LATER CHECKING
1386                    ;*  :  :  :  :  BGNDO
1387                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 1ST NIBBLE OF DATA
1388                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1389                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' BIT
1390                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1391                    ;*  :  :  :  :  :  ENDIF
1392                    ;*  :  :  :  :  :  DO UNTIL 'XFER' CLEARS OR TIMEOUT
1393                    ;*  :  :  :  :  ENDDO
1394                    ;*  :  :  :  :  BGNDO
1395                    ;*  :  :  :  :  :  CLOCK SYSTEM TO BEGIN 2ND NIBBLE OF DATA
1396                    ;*  :  :  :  :  :  IF > 128 CLOCKS
1397                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO SET 'XFER' FOR 2ND NIBBLE
1398                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1399                    ;*  :  :  :  :  :  ENDIF
1400                    ;*  :  :  :  :  :  DO UNTIL 'XFER' SETS OR TIMEOUT
1401                    ;*  :  :  :  :  ENDDO
1402                    ;*  :  :  :  :  GET PARITY BIT FROM INTSTA WORD FOR ASSEMBLED BYTE
1403                    ;*  :  :  :  :  BGNDO
1404                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 2ND NIBBLE
1405                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1406                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' ON 2ND NIBBLE
1407                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1408                    ;*  :  :  :  :  :  ENDIF
1409                    ;*  :  :  :  :  :  DO UNTIL 'XFER' IS CLEAR OR TIMEOUT
1410                    ;*  :  :  :  :  ENDDO
1411                    ;*  :  :  :  :  CLOCK SYSTEM TWICE MORE TO 'SET' INTERNAL STATUS
1412                    ;*  :  :  :  :  IF 8TH ASSEMBLED WRTDAT BYTE-ECC CHAR.
1413                    ;*  :  :  :  :  :  THEN-CLEAR CHARACTER COUNTER FOR ECC DETECTION
1414                    ;*  :  :  :  :  :  :  GET STORED WMCSTA FROM 1ST NIBBLE TIME
1415                    ;*  :  :  :  :  :  :  IF WMCSTA INDICATES 'ECC' BIT SET
1416                    ;*  :  :  :  :  :  :  :  THEN-EXIT 'DO' LOOP
1417                    ;*  :  :  :  :  :  :  :  ELSE-REPORT ASSEM. BYTE FROM WMC WAS INCORRECT
1418                    ;*  :  :  :  :  :  :  ENDIF
1419                    ;*  :  :  :  :  :  ELSE-COMPARE ACTUAL BYTE IN WRTDAT WITH EXPECTED BYTE
1420                    ;*  :  :  :  :  :  :  IF SAME
1421                    ;*  :  :  :  :  :  :  :  THEN-CONTINUE
1422                    ;*  :  :  :  :  :  :  :  ELSE-REPORT ASSEM. BYTE FROM WMC WAS INCORRECT
1423                    ;*  :  :  :  :  :  :  ENDIF
1424                    ;*  :  :  :  :  :  :  IF PARITY ON ASSEMBLED BYTE IS CORRECT
1425                    ;*  :  :  :  :  :  :  :  THEN-CONTINUE
1426                    ;*  :  :  :  :  :  :  :  ELSE-REPORT BYTE PARITY ERROR ON ASSEM. BYTE
```

```
1427                                ;*  :  :  :  :  :  :  ENDIF
1428                                ;*  :  :  :  :  :  ENDIF
1429                                ;*  :  :  :  :  ENDIF
1430                                ;*  :  :  :  ENDIF
1431                                ;*  :  :  ENDIF
1432                                ;*  :  :  DO UNTIL ALL BYTES ASSEMBLED AND CHECKED FOR CURRENT PACKING MODE
1433                                ;*  :  ENDDO
1434                                ;*  :  CHECK BYTE COUNTER TO SEE IF=0
1435                                ;*  :  IF BYTE COUNTER=0
1436                                ;*  :  :  THEN-UPDATE FORMAT NUMBER FOR NEXT PASS
1437                                ;*  :  :  ELSE-REPORT FAILURE OF BYTE COUNTER TO GO ZERO WHEN EXPECTED
1438                                ;*  :  ENDIF
1439                                ;*  :  DO UNTIL FORMAT NUMBER-7 (ILLEGAL #)
1440                                ;*  ENDDO
1441                                ;*ENDTST
1442  4300                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1443                                ;*WMC3 MICRO TEST 01
1444                                ;*WMC3 MICRO ERROR 01
1445                                ;*WMC3-WMC BYTE ASSY TESTS
1446                                ;*M8959
1447                                ;*WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1448                                ;*BYTE-SCLK COUNT = LLL
1449                                ;*DATA FORMAT = MM
1450                                ;*SKIP COUNT  = NN
1451                                ;*TEST ABORTED!
1452                                ;*
1453                                ;*WMC3 MICRO TEST 01
1454                                ;*WMC3 MICRO ERROR 02
1455                                ;*WMC3-WMC BYTE ASSY TESTS
1456                                ;*M8959
1457                                ;*WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1458                                ;*TEST ABORTED!
1459                                ;*
1460                                ;*WMC3 MICRO TEST 01
1461                                ;*WMC3 MICRO ERROR 03
1462                                ;*WMC3-WMC BYTE ASSY TESTS
1463                                ;*M8959
1464                                ;*WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1465                                ;*TEST ABORTED!
1466                                ;*
1467                                ;*WMC3 MICRO TEST 01
1468                                ;*WMC3 MICRO ERROR 04
1469                                ;*WMC3-WMC BYTE ASSY TESTS
1470                                ;*M8959
1471                                ;*WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1472                                ;*
1473                                ;*WMC3 MICRO TEST 01
1474                                ;*WMC3 MICRO ERROR 05
1475                                ;*WMC3-WMC BYTE ASSY TESTS
1476                                ;*M8959
1477                                ;*WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
```

```
1478                    ;*BYTE-SCLK COUNT = LLL
1479                    ;*DATA FORMAT = MM
1480                    ;*SKIP COUNT  = NN
1481                    ;*TEST ABORTED!
1482                    ;*
1483                    ;*WMC3 MICRO TEST 01
1484                    ;*WMC3 MICRO ERROR 06
1485                    ;*WMC3-WMC BYTE ASSY TESTS
1486                    ;*M8959
1487                    ;*WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1488                    ;*BYTE-SCLK COUNT = LLL
1489                    ;*DATA FORMAT = MM
1490                    ;*SKIP COUNT  = NN
1491                    ;*TEST ABORTED!
1492                    ;*
1493                    ;*WMC3 MICRO TEST 01
1494                    ;*WMC3 MICRO ERROR 07
1495                    ;*WMC3-WMC BYTE ASSY TESTS
1496                    ;*M8959
1497                    ;*WMC BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
1498                    ;*BYTE-SCLK COUNT = LLL
1499                    ;*DATA FORMAT = MM
1500                    ;*SKIP COUNT  = NN
1501                    ;*
1502                    ;*WMC3 MICRO TEST 01
1503                    ;*WMC3 MICRO ERROR 10
1504                    ;*WMC3-WMC BYTE ASSY TESTS
1505                    ;*M8959
1506                    ;*WMC PARITY BIT ON ASSEMBLED DATA SHOULD -0 (ODD PARITY) AND WAS  1
1507                    ;*
1508                    ;*WMC3 MICRO TEST 01
1509                    ;*WMC3 MICRO ERROR 11
1510                    ;*WMC3-WMC BYTE ASSY TESTS
1511                    ;*M8959
1512                    ;*WMC PARITY BIT ON ASSEMBLED DATA WAS =0 AND SHOULD BE =1
1513                    ;*
1514                    ;*WMC3 MICRO TEST 01
1515                    ;*WMC3 MICRO ERROR 12
1516                    ;*WMC3-WMC BYTE ASSY TESTS
1517                    ;*M8959
1518                    ;*WMC BYTE COUNTER EXPECTED TO BE -0 - ALL BYTES RECEIVED FROM WMC
1519                    ;*BYTE-SCLK COUNT = LLL
1520                    ;*DATA FORMAT = MM
1521                    ;*SKIP COUNT  = NN
1522                    ;*
1523                    ;*WMC3 MICRO TEST 01
1524                    ;*WMC3 MICRO ERROR 13
1525                    ;*WMC3-WMC BYTE ASSY TESTS
1526                    ;*M8959
1527                    ;*WMC FAILED TO SET 'ECC' IN WMC STATUS ON 8TH 'XFER' CYCLE
1528                    ;*ECC CHAR. NOT PRESENT ON WMC OUTPUT
1529                    ;*BYTE-SCLK COUNT = LLL
1530                    ;*DATA FORMAT = MM
1531                    ;*SKIP COUNT  = NN
```

```
1532                                    ;*
1533                                    ;*WMC3 MICRO TEST 01
1534                                    ;*WMC3 MICRO ERROR 14
1535                                    ;*WMC3-WMC BYTE ASSY TESTS
1536                                    ;*M8959
1537                                    ;*WMC FAILED TO SET 'WMC LEFT' BIT DURING 10 PACKING MODE
1538                                    ;*DATA FORMAT = MM
1539                                    ;*SKIP COUNT  = NN
1540  4300                              S
 (1)                                    ; ***************************************************************************
1541
1542  4300                              TST1:   TESTX   @1
 (1)  4300  3E  01            7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1543                                    ;%WMC3-WMC BYTE ASSY TESTS
1544                                    ;&M8959
1545
1546                                    ;INIT FORMAT CODE COUNTER TO 0 - 1ST FORMAT CODE = 11 NORMAL
1547
1548  4305  AF                    4.0  TST1ST: XRA     A
1549  4306  32  E4  45            13.0         STA     FORMT           ;SAVE FORMAT CODE
1550  4309  32  E3  45            13.0         STA     EABCNT          ;CLEAR THE LOOP CONTROL
1551                                    ;INIT COUNTER CONTROL FOR BYTE COUNTER AND ECODE COUNTER
1552
1553                                    ;RESTART WRITE MICROCONTROLLER ROM PROGRAM
1554
1555  430C  3E  FF                7.0  TST1L:  MVI     A,@377          ;-1 TO OFFSET NUMBER
1556  430E  32  E5  45            13.0         STA     DPNUM           ;INIT DATA PATTERN OFFSET SO 1ST WILL BE '0''
1557  4311  3E  01                7.0         MVI     A,W.RST         ;GET 'RST' BIT
1558  4313  D3  D3                10.0         OUT     WMCCTL          ;RESTART WMC
1559  4315  3E  05                7.0         MVI     A,SSTEP         ;SET THE SYSTEM TO SINGLE STEP MODE
1560  4317  32  DE  45            13.0         STA     CCTLWD          ;SAVE THE CONTROL WORD
1561  431A  D3  F0                10.0         OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
1562  431C  CD  46  48            18.0         CALL    CLKSYS          ;CLOCK THE WRITE PATH
1563  431F  CD  46  48            18.0         CALL    CLKSYS          ;CLOCK THE WRITE PATH
1564  4322  AF                    4.0         XRA     A               ;FINISH THE WMC RESTART
1565  4323  D3  D3                10.0         OUT     WMCCTL
1566  4325  3E  30                7.0         MVI     A,@60
1567  4327  D3  D7                10.0         OUT     CNTCTL          ;SELECT BYTE COUNTER
1568  4329  3E  B0                7.0         MVI     A,@260
1569  432B  D3  D7                10.0         OUT     CNTCTL          ;SELECT ECODE COUNTER
1570
1571                                    ;CLEAR COUNTERS
1572
1573  432D                              CLRBCT                          ;CLEAR BYTE COUNTER
 (1)  432D  AF                    4.0                 XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  432E  D3  D4                10.0                OUT     BYTCNT          ;CLEAR BITS 7-0
 (1)  4330  D3  D4                10.0                OUT     BYTCNT          ;CLEAR BITS 15-8
1574  4332                              CLRECT                          ;CLEAR ECODE COUNTER
 (1)  4332  AF                    4.0                 XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4333  D3  D6                10.0                OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4335  D3  D6                10.0                OUT     ERRCNT          ;CLEAR BITS 15-8
1575
1576  4337  32  E9  45            13.0         STA     ECCCNT          ;CLEAR 'ECC' CHECK COUNTER
```

```
1577  433A  32  EA  45   13.0          STA   CLKCNT          ;CLEAR THE 'SCLK' COUNTER
1578  433D  32  E1  45   13.0          STA   ABCNT           ;CLEAR THE ASSEM. BYTE COUNTER
1579  4340  32  E2  45   13.0          STA   ABCNTR
1580
1581                                ;SET DDR CONTROL TO 'OUT' FOR DATA INJECTION INTO BAL LOGIC
1582
1583  4343  3E  88       7.0          MVI   A,@210          ;SET 'OUT' BITS
1584  4345  D3  DB      10.0          OUT   DDRCTL          ;LOAD DDR CONTROL
1585
1586  4347  CD  F8  45  18.0          CALL  GENDAT          ;GENERATE THE 1ST SET OF DATA
1587  434A  CD  D3  46  18.0          CALL  GCDAT           ;AND THE 1ST SET OF BAL COMPARE BYTES
1588  434D  3E  04       7.0          MVI   A,4             ;GET THE SCLK COUNT FOR RELOAD
1589  434F  32  EB  45   13.0          STA   SCCNTR          ;SAVE IT
1590                                ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1591
1592  4352  3A  E4  45   13.0          LDA   FORMT           ;GET THE FORMAT CODE
1593  4355  CD  03  47   18.0          CALL  SHL4            ;POSITION IT TO BITS 6-4
1594  4358  D3  D0       10.0          OUT   DATACTL         ;LOAD DATA CONTROL WORD
1595  435A                             ROUT  R02H            ;ALSO SAVE IN CAS REG 02
 (1)  435A  D3  85       10.0                OUT   R02H              ;WRITE AC INTO R02H
 (1)  435C  7F           4.0                 MOV   A,A               ;RETRY LINK
1596  435D  3E  04       7.0          MVI   A,4             ;SET RECORD COUNT TO 1
1597  435F                             ROUT  R02L
 (1)  435F  D3  84       10.0                OUT   R02L              ;WRITE AC INTO R02L
 (1)  4361  7F           4.0                 MOV   A,A               ;RETRY LINK
1598  4362  06  B4       7.0          MVI   B,180           ;CLOCK THE SYSTEM THRU THE RESTART AREA
1599  4364  CD  46  48   18.0  T1RST:  CALL  CLKSYS          ;CLOCK THE SYSTEM
1600  4367  05           4.0          DCR   B
1601  4368  C2  64  43   10.0          JNZ   T1RST           ;STAY HERE TILL DONE COUNTING
1602  436B  3A  DF  45   13.0          LDA   DCNTWD          ;GET COUNT OF BYTES/4 SCLK
1603  436E  CD  63  47   18.0          CALL  LDBCNT          ;LOAD THE BYTE COUNTER FOR THIS FMT MODE
1604
1605                                ;ENABLE WRITE MICROCONTROLLER - START OF DATA XFR - WRITE
1606
1607  4371  3E  88       7.0  TST1A:  MVI   A,W.ENAB.W.WRITE  ;SELECT 'WMC EN' + 'WRITE' (PE MODE)
1608  4373  D3  D3      10.0          OUT   WMCCTL          ;LOAD WRITE MICROCONTROLLER CONTROL
1609  4375  AF           4.0          XRA   A
1610  4376  32  E8  45   13.0          STA   CLKON           ;CLEAR 'SCLK' ON FLAG
1611
1612  4379  06  FF       7.0          MVI   B,@377          ;SET SINGLE STEP CLOCK 'TIMEOUT' LIMIT TO MAX POS. COUNT
1613  437B  48           4.0          MOV   C,B             ;COPY FOR THE 'XFER' TIMEOUT COUNTER
1614
1615  437C  CD  1A  44   18.0  TST1B:  CALL  CKSCLK          ;ISSUE SINGLE STEP CLOCK
1616  437F  DA  0C  43   10.0          JC    TST1L           ;JUMP BACK IF ERROR DETECTED
1617  4382  DB  D0       10.0          IN    WMCSTA          ;GET WMC STATUS WORD
1618  4384  E6  20       7.0          ANI   W.XFER          ;SAVE ONLY 'XFER' BIT
1619  4386  C2  2D  44   10.0          JNZ   XFRBIT          ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
1620  4389  3A  F6  45   13.0          LDA   SXFER           ;GET THE SOFTWARE XFER FLAG
1621  438C  A7           4.0          ANA   A
1622  438D  CA  9A  43   10.0          JZ    TST1C           ;JUMP IF XFER NOT SET
1623  4390  AF           4.0          XRA   A               ;CLEAR SOFTWARE FLAG
1624  4391  32  F6  45   13.0          STA   SXFER           ;FOR NEXT PASS
1625  4394  C5          12.0          PUSH  B
1626  4395  06  7F       7.0          MVI   B,@177          ;SETUP A WATCHDOG COUNT
```

```
1627  4397  C3  53  44  10.0          JMP     XFRB2           ;PROCESS THE 'XFER' BIT DETECTED
1628                              ;HERE TO SEE IF 'SCLK' TIMEOUT HAPPENED
1629
1630  439A  05                4.0   TST1C:  DCR     B               ;COUNT CLOCK CYCLE
1631  439B  C2  AC  43  10.0          JNZ     TST1D           ;TIMEOUT? - JUMP IF OK
1632
1633  439E                            ERR     TST1L,TST1BC,3
 (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  439E  CD  09  28  18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43A1  01                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43A2  03                                  .BYTE   3
 (1)  43A3  CD  15  28  18.0            TST1BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43A6  DA  0C  43  10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1634                              ;>WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1635                              ;<TEST ABORTED!
1636  43A9  C3  18  28  10.0          JMP     TSTEND          ;TERMINATE TESTING NOW! - FATAL ERROR
1637
1638                              ;HERE TO SEE IF 'XFER' BIT TIMEOUT HAPPENED
1639
1640  43AC  0D                4.0   TST1D:  DCR     C               ;DECREMENT 'XFER' WATCHDOG COUNTER
1641  43AD  C2  7C  43  10.0          JNZ     TST1B           ;BACK TO LOOP IF OK
1642
1643  43B0                            ERR     TST1L,TST1DC
 (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43B0  CD  09  28  18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43B3  02                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43B4  00                                  .BYTE
 (1)  43B5  CD  15  28  18.0            TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43B8  DA  0C  43  10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1644                              ;>WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1645                              ;<TEST ABORTED.
1646  43BB  C3  18  28  10.0          JMP     TSTEND
```

```
1648  43BE                              S
 (1)                                    ;*****************************************************************************
1649                                    ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
1650                                    ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
1651  43BE                              S
 (1)                                    ;*****************************************************************************
1652
1653  43BE  3A  E8  45      13.0  SCLKBIT: LDA     CLKON           ;SEE IF 'SCLK' WAS ON
1654  43C1  A7                4.0         ANA     A               ;SET CONDITION BITS
1655  43C2  C0               12.0         RNZ                     ;EXIT IF CLOCK WAS SET-HAVE DATA ALREADY
1656  43C3  CD  27  46       18.0         CALL    SETDDR          ;THEN LOAD THE DATA INTO THE DDR REGS
1657  43C6  C5               12.0         PUSH    B               ;SAVE REG B & C
1658  43C7  06  14            7.0         MVI     B,20            ;GET A WATCHDOG COUNT
1659  43C9  CD  46  48       18.0  SCKWT:  CALL    CLKSYS          ;CLOCK THE SYSTEM - WAIT FOR 'SCLK' TO GO AWAY
1660  43CC  DB  D0           10.0         IN      WMCSTA          ;GET THE WMC STATUS WORD
1661  43CE  E6  20            7.0         ANI     W.XFER          ;DID 'XFER' BIT SET?
1662  43D0  CA  D6  43       10.0         JZ      SCKWT1          ;NO - JUMP OVER
1663  43D3  32  F6  45       13.0         STA     SXFER           ;YUP - SET THE FLAG
1664  43D6  DB  C0           10.0  SCKWT1: IN      DBUSSTA         ;GET THE DATA BUS STATUS INFO
1665  43D8  E6  02            7.0         ANI     T.SCLK          ;SEE IF SCLK IS GONE YET
1666  43DA  CA  EF  43       10.0         JZ      SCKGON          ;YUP - PROCEED
1667  43DD  05                4.0         DCR     B               ;NOPE! - DOWNCOUNT WATCHDOG
1668  43DE  C2  C9  43       10.0         JNZ     SCKWT           ;STAY IN LOOP TILL NO 'SCLK' OR DIE
1669  43E1                               ERR     SCKWTE,SCKWTC
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43E1  CD  09  28       18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0003                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E4  03                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E5  00                          .BYTE
 (1)  43E6  CD  15  28       18.0  SCKWTC::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E9  DA  19  44       10.0                   JC      SCKWTE          ;LOOP ADDRESS IF LOOP SPECIFIED
1670                                    ;>WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1671                                    ;<TEST ABORTED!
1672  43EC  C3  18  28       10.0         JMP     TSTEND          ;EXIT IF NOT LOOPING ON ERROR
1673  43EF  C1               10.0  SCKGON: POP     B               ;ALL OK - NOW RESET REG B & C
1674  43F0  3A  FB  45       13.0         LDA     SCCNTR          ;GET COUNTDOWN TO RELOAD DDR DATA
1675  43F3  3D                4.0         DCR     A
1676  43F4  32  EB  45       13.0         STA     SCCNTR          ;SAVE THE UPDATED COUNT
1677  43F7  C2  02  44       10.0         JNZ     SCKB1           ;JMP IF NOT TIME TO RELOAD DDR
1678
1679  43FA  CD  F8  45       18.0         CALL    GENDAT          ;GENERATE NEW DDR DATA FOR INJECTION
1680  43FD  3E  04            7.0         MVI     A,4             ;RESET THE COUNTER
1681  43FF  32  EB  45       13.0         STA     SCCNTR          ;FOR THE NEXT SET OF 'SCLK' CYCLES
1682
1683  4402  06  7F            7.0  SCKB1:  MVI     B,@177          ;GET NEW TIMEOUT COUNT
1684  4404  78                4.0         MOV     A,B             ;COPY TO REG A
1685  4405  32  E8  45       13.0         STA     CLKON           ;SET SCLK 'ON' FLAG
1686  4408  3A  EA  45       13.0         LDA     CLKCNT          ;GET THE CURRENT SCLK COUNT
1687  440B  3C                4.0         INR     A               ;UPDATE IT
1688  440C  32  EA  45       13.0         STA     CLKCNT          ;SAVE THE NEW COUNT
1689
1690  440F  CD  A4  45       18.0         CALL    LFTCHK          ;SEE IF 'LEFT' CHECK IS NECESSARY
1691  4412  3A  E6  45       13.0         LDA     LFTTST          ;SEE IF HAVE TO TEST FOR LOADING DDR IN 'LEFT' MODE
1692  4415  A7                4.0         ANA     A               ;SET THE CONDITION BITS
```

```
1693  4416  C4  C3  45      18.0          CNZ     LFTCMP          ;COMPLIMENT THE 'LFTFLG' FLAG
1694  4419  C9              10.0  SCKWTE: RET
1695  441A                        S
 (1)                                ; ***************************************************************************
1696                                ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
1697                                ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
1698                                ;       THE DDR.
1699  441A                          S
 (1)                                ; ***************************************************************************
1700
1701  441A  CD  46  48      18.0  CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
1702  441D  DB  C0          10.0          IN      DBUSSTA         ;GET STATUS OF DATA BUS
1703  441F  E6  02           7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
1704  4421  C2  29  44      10.0          JNZ     CKSC1           ;JUMP IF SCLK IS ON
1705  4424  AF               4.0          XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
1706  4425  32  E8  45      13.0          STA     CLKON
1707  4428  C9              10.0          RET                     ;EXIT
1708
1709  4429  CD  BE  43      18.0  CKSC1:  CALL    SCLKBIT         ;LOAD NEW DATA INTO DDR REGS
1710  442C  C9              10.0          RET                     ;EXIT
```

```
1712  442D                              S
 (1)                                    ;*****************************************************************************
1713                                    ;XFRBIT -- HERE WHEN WMC STATUS INDICATES 'XFER' BIT IS SET. ISSUE 2
1714                                    ;       MORE CLOCKS THEN CHECK WRITE DATA WORD - THIS IS OUTPUT OF BAL
1715                                    ;       AND SHOULD COMPARE TO BYTE STORED IN 'BALDAT' COMPARE BUFFER.
1716                                    ;       REG 'D' + 'E' CONTAIN POINTER TO EXPECTED DATA BYTE.  REG 'C'
1717                                    ;       CONTAINS THE NUMBER OF BYTES TO CHECK FOR EACH 4 SCLK CYCLE.
1718  442D                              S
 (1)                                    ;*****************************************************************************
1719
1720  442D  C5              12.0  XFRBIT: PUSH    B               ;SAVE REG B
1721  442E  DB  D0          10.0          IN      WMCSTA          ;GET THE CURRENT WMC STATUS
1722  4430  32  F5  45      13.0          STA     SWMCSTA         ;SAVE IT FOR THE ECC CHECK (LATER)
1723  4433  06  14           7.0          MVI     B,20            ;ALLOW UP TO 20. WMC CLOCKS TO DROP XFER
1724  4435  CD  1A  44      18.0  XFRB1:  CALL    CKSCLK          ;NEED TWO MORE CLOCK CYCLES BEFORE
1725  4438  DA  0C  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
1726  443B  DB  D0          10.0          IN      WMCSTA          ;GET WMC STATUS
1727  443D  E6  20           7.0          ANI     W.XFER          ;SAVE THE XFER BIT
1728  443F  CA  4B  44      10.0          JZ      XFRGON          ;OK TO CONTINUE IF XFER IS GONE
1729  4442  05               4.0          DCR     B               ;DECREMENT WATCHDOG COUNTER
1730  4443  C2  35  44      10.0          JNZ     XFRB1           ;LOOP TIL =0 OR XFER IS GONE
1731  4446                                ERR     TST1L,XFRGON
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4446  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0004                         MSGN            MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4449  04                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  444A  00                         .BYTE
 (1)  444B  CD  15  28      18.0  XFRGON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  444E  DA  0C  43      10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1732                                    ;>WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1733                                    ;HERE TO LOOK FOR 'XFER' TO SET FOR THE 2ND NIBBLE OF THIS BYTE
1734
1735  4451  06  7F           7.0          MVI     B,@177
1736  4453  CD  1A  44      18.0  XFRB2:  CALL    CKSCLK          ;SEE IF SCLK IS SET (ALSO CLOCK)
1737  4456  DA  0C  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
1738  4459  DB  D0          10.0          IN      WMCSTA          ;GET WMC STATUS
1739  445B  E6  20           7.0          ANI     W.XFER          ;SAVE ONLY XFER BIT
1740  445D  C2  72  44      10.0          JNZ     XFRB3           ;JUMP IF XFER IS SET - OK
1741  4460  05               4.0          DCR     B               ;LOOK FOR TIMEOUT
1742  4461  C2  53  44      10.0          JNZ     XFRB2           ;GO BACK TO LOOP IF OK
1743  4464                                ERR     TST1L,XFRB2C,3
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4464  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4467  05                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4468  03                         .BYTE   3
 (1)  4469  CD  15  28      18.0  XFRB2C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  446C  DA  0C  43      10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1744                                    ;>WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1745                                    ;<TEST ABORTED!
1746  446F  C3  18  28      10.0          JMP     TSTEND
1747
1748                                    ;HERE TO LOOK FOR 'XFER' BIT TO GO AWAY TO INDICATE FULL BYTE IS AVAIL.
1749                                    ;FROM THE 'WRTDAT' WORD
```

```
1750
1751  4472  06 14        7.0  XFRB3:  MVI   B,20
1752  4474  DB E0       10.0          IN    INTSTA          ;GET THE DATA PARITY BIT
1753  4476  32 F7 45    13.0          STA   SINTSTA         ;SAVE FOR PARITY CHECK
1754  4479  CD 1A 44    18.0  XFRB3A: CALL  CKSCLK          ;CLOCK THE SYSTEM TILL XFER BIT GOES AWAY
1755  447C  DA 0C 43    10.0          JC    TST1L           ;JUMP IF ERROR DETECTED
1756  447F  DB D0       10.0          IN    WMCSTA          ;GET THE WMC STATUS
1757  4481  E6 20        7.0          ANI   W.XFER          ;SAVE ONLY THE XFER BIT
1758  4483  CA 98 44    10.0          JZ    XFRB4           ;JUMP IF GONE - OK
1759  4486  05           4.0          DCR   B               ;DOWNCOUNT WATCHDOG
1760  4487  C2 79 44    10.0          JNZ   XFRB3A          ;BACK TO LOOP IF OK
1761  448A                            ERR   TST1L,XFRB3C,3
 (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  448A  CD 09 28    18.0                CALL  ERLP              ;PROCESS ERROR - DO 2.3
 (1)       0006                     MSGN  -     MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448D  06                       .BYTE MSGN                    ;MESSAGE NUMBER ID
 (1)  448E  03                       .BYTE 3
 (1)  448F  CD 15 28    18.0  XFRB3C::       CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4492  DA 0C 43    10.0                JC    TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1762                            ;>WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1763                            ;<TEST ABORTED.
1764  4495  C3 18 28    10.0          JMP   TSTEND
1765                            ;HERE TO TEST THE ASSEMBLED BYTE DATA AGAINST THE EXPECTED BYTE DATA
1766
1767  4498  CD 46 48    18.0  XFRB4:  CALL  CLKSYS          ;ALLOW THE BYTE COUNTER TO TICK
1768  449B  CD 46 48    18.0          CALL  CLKSYS          ;NEED 2 CLOCKS FOR 1 WMC INSTR.
1769  449E  C1          10.0          POP   B               ;RESTORE REG B
1770  449F  3A E9 45    13.0          LDA   ECCCNT          ;GET 'XFER' COUNT
1771  44A2  3C           4.0          INR   A               ;ADD 1
1772  44A3  32 E9 45    13.0          STA   ECCCNT          ;SAVE THE UPDATED COUNT
1773  44A6  FE 08        7.0          CPI   8               ;TIME TO CHECK 'ECC' BIT?
1774  44A8  CA 87 45    10.0          JZ    ECCCHK          ;TEST 'ECC' IF COUNT-8
1775
1776                            ;GET ASSEMBLED BAL DATA BYTE FROM WRITE DATA WORDS
1777
1778  44AB  3A E1 45    13.0          LDA   ABCNT           ;GET THE BYTE # UNDER TEST
1779  44AE  3C           4.0          INR   A               ;ADD 1
1780  44AF  32 E1 45    13.0          STA   ABCNT           ;SAVE THE BYTE NUMBER - UPDATED
1781  44B2                            ROUT  R05L
 (1)  44B2  D3 8A       10.0                OUT   R05L              ;WRITE AC INTO R05L
 (1)  44B4  7F           4.0                MOV   A,A               ;RETRY LINK
1782  44B5  AF           4.0          XRA   A
1783  44B6                            ROUT  R05H
 (1)  44B6  D3 88       10.0                OUT   R05H              ;WRITE AC INTO R05H
 (1)  44B8  7F           4.0                MOV   A,A               ;RETRY LINK
1784  44B9  EB           4.0          XCHG                  ;PUT 'EXPECTED' POINTER IN H & L
1785  44BA  7E           7.0          MOV   A,M             ;GET CORRECT BAL BYTE
1786  44BB                            ROUT  EDATA           ;SAVE 'EXPECTED' DATA
 (1)  44BB  D3 95       10.0                OUT   EDATA             ;WRITE AC INTO EDATA
 (1)  44BD  7F           4.0                MOV   A,A               ;RETRY LINK
1787  44BE  DB D3       10.0          IN    WRTDAT          ;GET ASSEMBLED DATA BYTE
1788  44C0                            ROUT  ADATA           ;SAVE 'ACTUAL' DATA
 (1)  44C0  D3 94       10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  44C2  7F           4.0                MOV   A,A               ;RETRY LINK
```

```
1789  44C3  BE                    7.0          CMP     M                ;EXPECTED = ACTUAL?
1790  44C4  EB                    4.0        . XCHG                     ;RESET H & L REGS
1791  44C5  13                    6.0          INX     D                ;POINT TO NEXT 'EXPECTED' ENTRY
1792  44C6  CA    D4    44       10.0          JZ      TST1XC           ;JUMP OVER ERROR IF DATA BYTE OK
1793  44C9  3A    E1    45       13.0          LDA     ABCNT            ;GET THE COUNT OF ASSEM. BYTES AT FAILURE
1794  44CC  32    E3    45       13.0          STA     EABCNT           ;SAVE FOR THE LOOP CONTROL
1795  44CF                                    ERRB    DELOOP,TST1XC,3
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44CF  CD    12    28       18.0                    CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0007                                         MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D2  07                                           .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44D3  03                                           .BYTE   3                ;PRINT ROUTINE NUMBER
 (1)  44D4  CD    15    28       18.0          TST1XC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D7  DA    6C    45       10.0                    JC      DELOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1796                                     ;>WMC BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
1797                                     ;HERE TO CHECK THE PARITY BIT ON THE DATA ASSEMBLED
1798
1799  44DA  DB    D3              10.0  XFRB4P: IN      WRTDAT           ;GET THE DATA BYTE AGAIN
1800  44DC  A7                     4.0          ANA     A                ;SET THE CONDITION BITS
1801  44DD  E4    EC    44       18.0          CPO     DATPO            ;CHECK FOR ODD PARITY
1802  44E0  DA    0C    43       10.0          JC      TST1L            ;JUMP IF ERROR DETECTED
1803  44E3  EC    02    45       18.0          CPE     DATPE            ;CHECK FOR PARITY EVEN - BIT -1
1804  44E6  DA    0C    43       10.0          JC      TST1L            ;JUMP IF ERROR DETECTED
1805  44E9  C3    14    45       10.0          JMP     XFRB5            ;ON TO THE NEXT SECTION....
1806
1807                                     ;HERE TO CHECK FOR PARITY ODD - BIT =0
1808
1809  44EC  F5                     12.0  DATPO:  PUSH    PSW              ;SAVE THE PSW STATUS
1810  44ED  3A    F7    45       13.0          LDA     SINTSTA          ;GET THE PARITY BIT (IN BIT 0)
1811  44F0  E6    10               7.0          ANI     WDR.P            ;SAVE JUST THE PARITY BIT
1812  44F2  CA    00    45       10.0          JZ      DATPOE           ;EXIT IF OK
1813  44F5                                    ERR     DATPOE,DATPOC
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F5  CD    09    28       18.0                    CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0008                                         MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F8  08                                           .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44F9  00                                           .BYTE
 (1)  44FA  CD    15    28       18.0          DATPOC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FD  DA    00    45       10.0                    JC      DATPOE           ;LOOP ADDRESS IF LOOP SPECIFIED
1814                                     ;>WMC PARITY BIT ON ASSEMBLED DATA SHOULD =0 (ODD PARITY) AND WAS -1
1815
1816  4500  F1                     10.0  DATPOE: POP     PSW              ;RESET THE PSW STATUS
1817  4501  C9                     10.0          RET
1818
1819                                     ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
1820
1821  4502  3A    F7    45       13.0  DATPE:  LDA     SINTSTA          ;GET THE BIT
1822  4505  E6    10               7.0          ANI     WDR.P            ;SAVE JUST THE PARITY DATA BIT
1823  4507  C0                     12.0          RNZ
```

```
1825  4508                                        ERR     DATPEE,DATPEC
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4508  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0009                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4508  09                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450C  00                                           .BYTE
 (1)  450D  CD  15  28        18.0               DATPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4510  DA  13  45        10.0                        JC      DATPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
1826                                    ;>WMC PARITY BIT ON ASSEMBLED DATA WAS =0 AND SHOULD BF =1
1827  4513  C9                 10.0    DATPEE: RET
1828  4514  3A  E2  45         13.0    XFRB5:  LDA     ABCNTR          ;GET COUNT OF BYTES LEFT THIS CYCLE
1829  4517  3D                  4.0            DCR     A               ;DECREMENT # OF ASSEMBLED BYTES - COUNT
1830  4518  32  E2  45         13.0            STA     ABCNTR          ;SAVE IT
1831  451B  CC  D3  46         18.0            CZ      GCDAT           ;GENERATE NEW 'EXPECTED' BUFFER BASED ON
1832                                                                    ;CURRENT DDR DATA
1833  451E  06  7F               7.0           MVI     B,@177
1834  4520  48                  4.0            MOV     C,B             ;RESET WATCHDOG FOR 'XFER' BIT
1835  4521  E5                 12.0            PUSH    H
1836  4522  21  E0  45         10.0            LXI     H,IMBYTC        ;POINT TO THE AMT OF BYTES EXPECTED
1837  4525  3A  E1  45         13.0            LDA     ABCNT           ;THEN GET # BYTES ASSEMBLED BY THE WMC
1838  4528  BE                  7.0            CMP     M               ;SEE IF DONE
1839  4529  E1                 10.0            POP     H               ;RESET REG H & L
1840  452A  C2  9A  43         10.0            JNZ     TST1C           ;BACK TO LOOP IF MORE TO DO
1841
1842                                    ;DUMMY READ OF BYTE COUNTER
1843  452D  DB  D4               10.0            IN      BYTCNT
1844  452F  DB  D4               10.0            IN      BYTCNT
1845
1846  4531  DB  D4               10.0            IN      BYTCNT          ;GET BYTE COUNTER 7-0
1847  4533  47                  4.0            MOV     B,A             ;TEMP STORE
1848  4534  DB  D4               10.0            IN      BYTCNT          ;GET BYTE CNT 15-8
1849  4536  78                  4.0            MOV     A,B
1850  4537  A7                  4.0            ANA     A
1851  4538  CA  43  45         10.0            JZ      XFRB5C          ;JUMP IF BYTE COUNT OK
1852  4538                                    ROUT    ADATA           ;STORE THE ACTUAL BYTE COUNTER 7-0
 (1)  4538  D3  94               10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  453D  7F                   4.0                    MOV     A,A             ;RETRY LINK
1853  453E                                    ERRA    TST1L,XFRB5C,3
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  453E  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000A                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4541  0A                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4542  03                                           .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4543  CD  15  28        18.0               XFRB5C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4546  DA  0C  43        10.0                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1854                                    ;>WMC BYTE COUNTER EXPECTED TO BE =0 - ALL BYTES RECEIVED FROM WMC
1855
1856                                    ;HERE TO UPDATE 'FORMAT' COUNT TO NEXT CODE NUMBER
1857
1858  4549  3A  E4  45         13.0            LDA     FORMT           ;GET CURRENT #
1859  454C  3C                  4.0            INR     A               ;INCREMENT
1860  454D  32  E4  45         13.0            STA     FORMT           ;SAVE THE NEW FORMAT CODE NUMBER
1861  4550  FE  07               7.0           CPI     @7              ;DONE WITH THIS TEST?
1862  4552  C2  0C  43         10.0            JNZ     TST1L           ;LOOP BACK TO TEST THIS FORMAT MODE
```

```
:864  4555                                        ENDTST  TST1ST
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4555                                        REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4555  CD  06  28        18.0                          CALL    REQST
 (2)  4558  00                                              .BYTE                    ;DATA PATTERN NUMBER
 (2)  4559  00  00                                          .WORD                    ;SYSTEM '#' COUNT
 (2)  455B  00  00                                          .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455D  00                                              .BYTE                    ;DATA COMPARE FLAG IF =1
 (2)  455E  07                                              .BYTE   7                ;REQUEST CODE
 (1)  455F  3A  9A  4F        13.0                  LDA     ITERA            ;GET ITERATION COUNT
 (1)  4562  3D                 4.0                  DCR     A                ;DOWNCOUNT
 (1)  4563  32  9A  4F        13.0                  STA     ITERA            ;SAVE COUNT
 (1)  4566  F2  05  43        10.0                  JP      TST1ST           ;DO TEST UNTIL TILL = 0
1865  4569  C3  18  28        10.0                  JMP     TSTEND           ;ALL DONE
1866                                     ;HERE IS THE DATA COMPARE ERROR LOOP CONTROL ROUTINE
1867
1868                                     ;WILL ENABLE THE TEST TO LOOP AT THE SPECIFIED BYTE IN ERROR
1869
1870  456C  3A  94  4F        13.0  DELOOP: LDA     LPFLG            ;SEE IF LOOP FLAG STILL SET
1871  456F  A7                 4.0          ANA     A
1872  4570  C2  79  45        10.0          JNZ     DELP1            ;JUMP IF LOOPING ON ERROR
1873  4573  32  E3  45        13.0          STA     EABCNT           ;NOT LOOPING - CLEAR THE FLAG
1874  4576  C3  14  45        10.0          JMP     XFRB5            ;AND EXIT
1875
1876  4579  3A  E1  45        13.0  DELP1:  LDA     ABCNT            ;GET THE CURRENT COUNT OF ASSEMBLED BYTES
1877  457C  47                 4.0          MOV     B,A              ;TEMP STORE
1878  457D  3A  E3  45        13.0          LDA     EABCNT           ;GET THE BYTE COUNT AT THE DETECTED ERROR
1879  4580  B8                 4.0          CMP     B                ;SEE IF AT THE SAME BYTE COUNT
1880  4581  CA  0C  43        10.0          JZ      TST1L            ;THEN JUMP TO RESTART TEST IF SAME
1881  4584  C3  14  45        10.0          JMP     XFRB5            ;ELSE, CONTINUE WITH THE TEST
1882
1883                                     ;HERE TO THROW AWAY ECC CHARACTER GENERATED EVERY 8TH BYTE OUTPUT FROM
1884                                     ;WMC.
1885
1886  4587  AF                 4.0  ECCCHK: XRA     A                ;CLEAR REG A
1887  4588  32  E9  45        13.0          STA     ECCCNT           ;ZERO COUNT OF 'XFER' CYCLES
1888  458B  3A  F5  45        13.0          LDA     SWMCSTA          ;GET THE 'STORED' WMC STATUS
1889  458E                                 ROUT    ADATA            ;SAVE FOR ERROR PRINTOUT
 (1)  458E  D3  94             10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4590  7F                 4.0                   MOV     A,A              ;RETRY LINK
1890  4591  E6  10             7.0          ANI     W.ECC            ;SAVE 'ECC' BIT
1891  4593  C2  9A  43        10.0          JNZ     TST1C            ;JUMP BACK TO CLOCK LOOP IF OK ('ECC' BIT WAS ON)
1892  4596                                 ERRA    ECCON1,ECCON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4596  CD  0F  28        18.0                  CALL    ERLPA            ;PROCESS ERROR - DO 2.3
 (1)        0008                                    MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4599  08                                      .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  459A  03                                      .BYTE   3                ;PRINT ROUTINE NUMBER
 (1)  459B  CD  15  28        18.0          ECCON:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  459E  DA  A1  45        10.0                  JC      ECCON1           ;LOOP ADDRESS IF LOOP SPECIFIED
1893                                     ;>WMC FAILED TO SET 'ECC' IN WMC STATUS ON 8TH 'XFER' CYCLE
1894                                     ;>ECC CHAR. NOT PRESENT ON WMC OUTPUT
1895  45A1  C3  18  28        10.0  ECCON1: JMP     TSTEND
```

```
1897  45A4                                  S
 (1)                                        ; ***************************************************************************
1898                                        ;LFTCHK -- ROUTINE TO TEST 'WMC LEFT' WHEN IN ANY 10 PACKING MODE.  'WMC
1899                                        ;       LEFT' WILL BE 0 FOR 11 NORMAL, 15 NORMAL, AND IMAGE MODE.
1900  45A4                                  S
 (1)                                        ; ***************************************************************************
1901
1902  45A4  3A  E6  45      13.0  LFTCHK: LDA     LFTTST          ;GET CHECK FLAG
1903  45A7  A7                4.0          ANA     A               ;SET CONDITION BITS
1904  45A8  C8               12.0          RZ                      ;EXIT IF NO TEST NEEDED NOW
1905  45A9  3A  E7  45       13.0          LDA     LFTFLG          ;GET 'WMC LEFT' ON/OFF FLAG
1906  45AC  A7                4.0          ANA     A               ;CHECK TO SEE IF ITS OFF
1907  45AD  CA  CB  45       10.0          JZ      LFTOFF
1908
1909                                      ;HERE TO SEE IF ANY 10 XFR MODE HAS SET WMC LEFT STATUS
1910
1911  45B0  DB  A1           10.0  LFTON:  IN      CBUSSTA         ;GET STATUS
1912  45B2  E6  04            7.0          ANI     W.LEFT          ;SAVE ONLY 'WMC LEFT' BIT
1913  45B4  C2  BC  45       10.0          JNZ     LFTOK2          ;JUMP IF 'LEFT' IS ON (ITS EXPECTED)
1914  45B7                                 ERR     TST1L,LFTOK2,2
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45B7  CD  09  28       18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45BA  0C                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45BB  02                            .BYTE   2
 (1)  45BC  CD  15  28       18.0  LFTOK2::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45BF  DA  0C  43       10.0                     JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1915                                    ;>WMC FAILED TO SET 'WMC LEFT' BIT DURING 10 PACKING MODE
1916  45C2  C9               10.0          RET
1917
1918                                    ;HERE TO COMPLEMENT THE STATE OF THE LEFT MODE CHECK FLAG
1919
1920  45C3  3A  E7  45       13.0  LFTCMP: LDA     LFTFLG          ;GET CURRENT 'LEFT' FLAG
1921  45C6  2F                 4.0          CMA                     ;COMPLEMENT-NEXT SCLK 'LEFT' SHOULD BE OFF
1922  45C7  32  E7  45       13.0          STA     LFTFLG          ;SAVE IT
1923  45CA  C9               10.0          RET
1924                                    ;HERE TO CHECK THAT ANY 10 XFR MODE HAS CLEARED WMC LEFT FLAG
1925
1926  45CB  DB  A1           10.0  LFTOFF: IN      CBUSSTA         ;GET CBUS STATUS INFO
1927  45CD  E6  04            7.0          ANI     W.LEFT          ;SAVE ONLY THE WMC LEFT BIT
1928  45CF  CA  D7  45       10.0          JZ      LFTOK1          ;JUMP IF NOT ON - OK
1929  45D2                                 ERR     TST1L,LFTOK1,2
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45D2  CD  09  28       18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000D                          MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D5  0D                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45D6  02                            .BYTE   2
 (1)  45D7  CD  15  28       18.0  LFTOK1::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45DA  DA  0C  43       10.0                     JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1930                                    ;>WMC FAILED TO CLEAR 'WMC LEFT' BIT DURING 10 PACKING MODE
1931  45DD  C9               10.0          RET                     ;EXIT
```

```
1933                                ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE
1934
1935  45DE  00                      CCTLWD: .BYTE   0           ;IMAGE OF CLOCK CONTROL WORD
1936  45DF  00                      DCNTWD: .BYTE   0           ;COUNTER FOR TOTAL BYTES ASSEMBLED 4 SCLKS
1937  45E0  00                      IMBYTC: .BYTE   0           ;IMAGE OF BYTE COUNTER 7-0 BITS
1938  45E1  00                      ABCNT:  .BYTE   0           ;ASSEMBLED BYTE COUNTER
1939  45E2  00                      ABCNTR: .BYTE   0
1940  45E3  00                      EABCNT: .BYTE   0           ;BYTE COUNT AT A DETECTED ERROR
1941  45E4  00                      FORMT:  .BYTE   0           ;FORMAT CODE SELECTED FOR TEST
1942  45E5  00                      DPNUM:  .BYTE   0           ;DATA PATTERN OFFSET NUMBER (0 TO 17)
1943  45E6  00                      LFTTST: .BYTE   0           ;'WMC LEFT' TEST NEEDED FLAG
1944  45E7  00                      LFTFLG: .BYTE   0           ;'WMC LEFT' ON/OFF FLAG
1945  45E8  00                      CLKON:  .BYTE   0           ;'SCLK' ON FLAG-USED TO TELL WHEN TO FILL DDR REGS
1946  45E9  00                      ECCCNT: .BYTE   0           ;COUNT BY 7 FOR ECC CHARACTER CHECK
1947  45EA  00                      CLKCNT: .BYTE   0           ;COUNT OF SCLKS GENERATED
1948  45EB  00                      SCCNTR: .BYTE   0           ;COUNT TO RELOAD THE DDR REGS
1949  45EC  00                      SDDRA:  .BYTE   0           ;DDRA DATA ACTUALLY LOADED
1950  45ED  00                      SDDRB:  .BYTE   0           ;DDRB DATA ACTUALLY LOADED
1951  45EE  00                      SDDRC:  .BYTE   0           ;DDRC DATA ACTUALLY LOADED
1952
1953  45EF  00                      SDDRAT: .BYTE   0           ;DDRA DATA FROM PGM TABLE
1954  45F0  00                      SDDRBT: .BYTE   0           ;DDRB DATA FROM PGM TABLE
1955  45F1  00                      SDDRCT: .BYTE   0           ;DDRC DATA FROM PGM TABLE
1956
1957  45F2  00                      SDDRAC: .BYTE   0           ;COPY OF DDRA 'TEMP' DATA
1958  45F3  00                      SDDRBC: .BYTE   0           ;COPY OF DDRB 'TEMP
1959  45F4  00                      SDDRCC: .BYTE   0           ;COPY OF DDRC 'TEMP
1960  45F5  00                      SWMCSTA: .BYTE  0           ;COPY OF THE WMC STATUS AT 'XFER' BIT TIME
1961  45F6  00                      SXFER:  .BYTE   0           ;SOFTWARE 'XFER' FLAG
1962  45F7  00                      SINTSTA: .BYTE  0           ;SOFTWARE COPY OF CBUSSTA WORD
```

```
1964  45F8                              S
 (1)                                    ;*****************************************************************
1965                                    ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT 'SCLK' CYCLE.
1966                                    ;       DATA IS COADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
1967                                    ;       'SDDRCC' FOR REFERENCE.
1968  45F8                              S
 (1)                                    ;*****************************************************************
1969
1970  45F8  F5              12.0  GENDAT: PUSH   PSW          ;SAVE PSW + A
1971  45F9  C5              12.0        PUSH   B            ;SAVE B + C
1972  45FA  E5              12.0        PUSH   H            ;SAVE H + L
1973  45FB  3A  E5  45      13.0        LDA    DPNUM        ;GET THE LAST PATTERN # USED
1974  45FE  3C               4.0        INR    A            ;POINT TO NEXT
1975  45FF  32  E5  45      13.0        STA    DPNUM        ;SAVE NUMBER
1976  4602  47               4.0        MOV    B,A
1977
1978  4603  87               4.0        ADD    A
1979  4604  80               4.0        ADD    B            ;MULT BY 3 FOR OFFSET
1980  4605  4F               4.0        MOV    C,A
1981  4606  06  00            7.0        MVI    B,0          ;B + C HAVE OFFSET
1982  4608  21  88  46       10.0        LXI    H,CDATBL     ;POINT TO DATA TABLE
1983  460B  09               10.0        DAD    B            ;ADD IN OFFSET TO H + L
1984
1985  460C  7E                7.0        MOV    A,M          ;GET DATA BYTE FOR DDR 'A'
1986  460D  32  EF  45       13.0        STA    SDDRAT       ;SAVE DDR 'A' IMAGE
1987  4610  32  F2  45       13.0        STA    SDDRAC       ;LOAD DDR 'A' COPY
1988  4613  23                6.0        INX    H            ;POINT TO NEXT DATA BYTE
1989  4614  7E                7.0        MOV    A,M          ;GET IT
1990  4615  32  F0  45       13.0        STA    SDDRBT       ;SAVE DDR 'B' IMAGE
1991  4618  32  F3  45       13.0        STA    SDDRBC       ;SAVE DDRB COPY
1992  461B  23                6.0        INX    H            ;POINT TO NEXT DATA BYTE
1993  461C  7E                7.0        MOV    A,M          ;GET IT
1994  461D  32  F1  45       13.0        STA    SDDRCT       ;SAVE DDR 'C' IMAGE
1995  4620  32  F4  45       13.0        STA    SDDRCC       ;SAVE DDRC COPY
1996
1997  4623  E1               10.0  GENOUT: POP    H            ;RESTORE REGS
1998  4624  C1               10.0        POP    B
1999  4625  F1               10.0        POP    PSW
2000  4626  C9               10.0        RET
```

```
2002  4627                                    $
 (1)                                          ;*****************************************************************************
2003                                          ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
2004                                          ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
2005                                          ;       THE MASSBUS. WMC 'LEFT' FLAG SET INDICATES THAT THIS ROUTINE IS NOT
2006                                          ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
2007                                          ;       'LEFT' FLAG IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
2008                                          ;       DDR FORMAT.
2009  4627                                    $
 (1)                                          ;*****************************************************************************
2010
2011  4627  3A  E7  45      13.0  SETDDR: LDA     LFTFLG          ;GET THE SOFTWARE 'LEFT' MODE FLAG
2012  462A  A7                4.0          ANA     A               ;SET THE CONDITION BITS
2013  462B  C2  66  46        10.0         JNZ     SLFTDDR         ;WMC SET TO 'LEFT' MODE - NO TRANSLATION
2014
2015                                        ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR
2016
2017  462E  3A  EF  45      13.0          LDA     SDDRAT          ;GET DDRA TEMP
2018  4631  E6  3F            7.0         ANI     @77             ;STRIP 2 BITS OFF
2019  4633  07                4.0         RLC
2020  4634  07                4.0         RLC                     ;LEFT JUSTIFY
2021  4635  47                4.0         MOV     B,A             ;TEMP SAVE
2022  4636  3A  F1  45      13.0          LDA     SDDRCT          ;GET DDRC TEMP
2023  4639  E6  03            7.0         ANI     @3              ;SAVE 2 BITS
2024  463B  B0                4.0         ORA     B
2025  463C  32  EC  45      13.0          STA     SDDRA           ;SAVE DDRA DESIRED
2026
2027  463F  3A  F0  45      13.0          LDA     SDDRBT          ;GET DDRB TEMP
2028  4642  E6  3F            7.0         ANI     @77
2029  4644  07                4.0         RLC
2030  4645  07                4.0         RLC                     ;LEFT JUSTIFY
2031  4646  47                4.0         MOV     B,A
2032  4647  3A  EF  45      13.0          LDA     SDDRAT          ;GET DDRA AGAIN
2033  464A  E6  C0            7.0         ANI     @300            ;SAVE 2 BITS
2034  464C  07                4.0         RLC
2035  464D  07                4.0         RLC
2036  464E  B0                4.0         ORA     B
2037  464F  32  ED  45      13.0          STA     SDDRB           ;SAVE DDRB DATA
2038
2039  4652  3A  F1  45      13.0          LDA     SDDRCT          ;GET DDRC TEMP
2040  4655  E6  04            7.0         ANI     @4              ;SAVE THE PARITY BIT
2041  4657  47                4.0         MOV     B,A             ;TEMP STORE
2042  4658  3A  F0  45      13.0          LDA     SDDRBT          ;GET DDRB AGAIN
2043  465B  E6  C0            7.0         ANI     @300            ;SAVE 2 BITS
2044  465D  07                4.0         RLC
2045  465E  07                4.0         RLC                     ;RIGHT JUSTIFY
2046  465F  B0                4.0         ORA     B
2047  4660  32  EE  45      13.0          STA     SDDRC           ;SAVE DDRC DATA
2048  4663  C3  78  46        10.0         JMP     SDDCOM          ;COMMON EXIT
2049                                      ;HERE IF NO MODIFICATION OF DDR DATA NECESSARY
2050
2051  4666  3A  EF  45      13.0  SLFTDDR: LDA     SDDRAT          ;GET DDRA DATA
2052  4669  32  EC  45      13.0          STA     SDDRA           ;SAVE FOR OUTPUT
2053  466C  3A  F0  45      13.0          LDA     SDDRBT          ;GET TEMP DDRB DATA
```

SEQ 0121

```
2054  466F  32  ED  45   13.0          STA   SDDRB          ;SAVE IT
2055  4672  3A  F1  45   13.0          LDA   SDDRCT         ;GET DDRC TEMP DATA
2056  4675  32  EE  45   13.0          STA   SDDRC          ;SAVE FOR OUTPUT
2057
2058  4678  3A  EC  45   13.0  SDDCOM: LDA   SDDRA          ;GET DATA TO OUTPUT
2059  467B  D3  D8        10.0          OUT   DDRA
2060  467D  3A  ED  45   13.0          LDA   SDDRB
2061  4680  D3  D9        10.0          OUT   DDRB
2062  4682  3A  EE  45   13.0          LDA   SDDRC
2063  4685  D3  DA        10.0          OUT   DDRC
2064  4687  C9              10.0          RET                  ;EXIT
2065                                ;TABLE OF DATA TO USE FOR 'DDR' LOADING + SCLK REQUESTS
2066                                ;DATA APPEARS IN THE TABLE FOR WMC 'LEFT' XFR MODE...IF WMC IS IN
2067                                ;'RIGHT' MODE, THEN THE DATA IS SHIFTED TO THE LEFT TWO PLACES.
2068
2069  4688  45                    CDATBL: .BYTE  $45          ;HEX 12345
2070  4689  23                            .BYTE  $23
2071  468A  C1                            .BYTE  $1
2072  468B  56                            .BYTE  $56          ;HEX 23456
2073  468C  34                            .BYTE  $34
2074  468D  02                            .BYTE  $2
2075  468E  67                            .BYTE  $67          ;HEX 34567
2076  468F  45                            .BYTE  $45
2077  4690  03                            .BYTE  $3
2078  4691  78                            .BYTE  $78          ;HEX 05678
2079  4692  56                            .BYTE  $56
2080  4693  00                            .BYTE  $0
2081  4694  89                            .BYTE  $89          ;HEX 16789
2082  4695  67                            .BYTE  $67
2083  4696  01                            .BYTE  $1
2084  4697  9A                            .BYTE  $9A          ;HEX 2789A
2085  4698  78                            .BYTE  $78
2086  4699  02                            .BYTE  $2
2087  469A  AB                            .BYTE  $AB          ;HEX 389AB
2088  469B  89                            .BYTE  $89
2089  469C  03                            .BYTE  $3
2090  469D  BC                            .BYTE  $BC          ;HEX 90ABC
2091  469E  9A                            .BYTE  $9A
2092  469F  00                            .BYTE  $0
2093  46A0  CD                            .BYTE  $CD          ;HEX 1ABCD
2094  46A1  AB                            .BYTE  $AB
2095  46A2  01                            .BYTE  $1
2096  46A3  DE                            .BYTE  $DE          ;HEX 2BCDE
2097  46A4  BC                            .BYTE  $BC
2098  46A5  02                            .BYTE  $2
2099  46A6  EF                            .BYTE  $EF          ;HEX 3CDEF
2100  46A7  CD                            .BYTE  $CD
2101  46A8  03                            .BYTE  $3
2102  46A9  F0                            .BYTE  $F0          ;HEX 0DEF0
2103  46AA  DE                            .BYTE  $DE
2104  46AB  00                            .BYTE  $0
2105  46AC  01                            .BYTE  $01          ;HEX 1GF01
2106  46AD  EF                            .BYTE  $EF
2107  46AE  01                            .BYTE  $1
```

```
2108  46AF  12                          .BYTE   $12             ;HEX 2F012
2109  46B0  F0                          .BYTE   $F0
2110  46B1  02                          .BYTE   $2
2111  46B2  23                          .BYTE   $23             ;HEX 30123
2112  46B3  01                          .BYTE   $01
2113  46B4  03                          .BYTE   $3
2114  46B5  34                          .BYTE   $34             ;HEX 01234
2115  46B6  12                          .BYTE   $12
2116  46B7  00                          .BYTE   $0
2117  46B8  00                  DTEND:  .BYTE   $0              ;0 - END OF TABLE
2118  46B9  00                          .BYTE   $0
2119  46BA  00                          .BYTE   $0
2120
2121                            ;BAL 'EXPECTED' OUTPUT DATA
2122
2123  46BB  00                  BALDAT: .BYTE   0               ;TABLE OF EXPECTED DATA BYTES
2124  46BC  00                          .BYTE   0               ;OUTPUT FROM BYTE ASSEMBLY
2125  46BD  00                          .BYTE   0               ;LOGIC OF WMC
2126  46BE  00                          .BYTE   0
2127  46BF  00                          .BYTE   0
2128  46C0  00                          .BYTE   0
2129  46C1  00                          .BYTE   0
2130  46C2  00                          .BYTE   0
2131  46C3  00                          .BYTE   0
2132  46C4  00                          .BYTE   0
2133
2134                            ;TABLE OF EXPECTED NUMBER OF DATA BYTES OUTPUT FROM BAL FOR EVERY SECTS
2135
2136  46C5  0C    47            FMTTBL: .ADDR   F11N            ;0-11 NORMAL - 8 BYTES/4 CLOCKS
2137  46C7  14    47                    .ADDR   F15N            ;1-15 NORMAL - 8 BYTES/4 CLOCKS
2138  46C9  87    47                    .ADDR   F10C            ;2-10 COMPAT - 8 BYTES/4 CLOCKS
2139  46CB  76    47                    .ADDR   F10D            ;3-10 DUMP - 10 BYTES/4 CLOCKS
2140  46CD  F9    47                    .ADDR   F10HD           ;4-10 HIGH DEN COMPAT - 9 BYTES/4 CLOCKS
2141  46CF  34    47                    .ADDR   FIMAG           ;5-IMAGE MODE - 4 BYTES/4 CLOCKS
2142  46D1  F9    47                    .ADDR   F10HD           ;4-10 HIGH DEN DUMP - 9 BYTES/4 CLOCKS
```

```
2144  46D3                                  $
 (1)                                        ;*****************************************************************
2145                                        ;GCDAT -- ROUTINE TO GENERATE THE EXPECTED DATA TABLE OF OUTPUT DATA
2146                                        ;       FROM THE BAL.  THIS ROUTINE IS CALLED AFTER 4TH SCLK CYCLE COMPLETE.
2147  46D3                                  $
 (1)                                        ;*****************************************************************
2148
2149  46D3  F5              12.0  GCDAT:  PUSH  PSW             ;D + E NOW POINT TO STORAGE
2150  46D4  C5              12.0        PUSH  B
2151  46D5  E5              12.0        PUSH  H
2152  46D6  21  BB  46      10.0        LXI   H,BALDAT        ;POINT TO EXPECTED TABLE IN H + L
2153  46D9  EB               4.0        XCHG                  ;D + E NOW USED AS POINTERS
2154  46DA  21  C5  46      10.0        LXI   H,FMTTBL        ;POINT TO START OF DATA GEN ROUTINES
2155  46DD  3A  E4  45      13.0        LDA   FORMT           ;GET FMT CODE #
2156  46E0  07               4.0        RLC                   ;MULT BY 2 TO GET OFFSET
2157  46E1  4F               4.0        MOV   C,A
2158  46E2  06  0C           7.0        MVI   B,0
2159  46E4  09              10.0        DAD   B               ;H + L NOW POINT TO GEN ROUTINE
2160  46E5  D5              12.0        PUSH  D               ;SAVE POINTER TO TABLE
2161  46E6  5E               7.0        MOV   E,M             ;GET ROUTINE LOW ADDR
2162  46E7  23               6.0        INX   H               ;POINT TO HIGH BYTE ADDRESS OF ROUTINE
2163  46E8  56               7.0        MOV   D,M             ;GET THE HIGH BYTE ADDRESS
2164  46E9  EB               4.0        XCHG                  ;MAKE H & L POINT TO ROUTINE
2165  46EA  D1              10.0        POP   D               ;RESET D & E
2166  46EB  E9               6.0        PCHL                  ;GO TO PROPER GEN ROUTINE
2167
2168  46EC  21  BB  46      10.0  GCDATX: LXI   H,BALDAT       ;POINT TO START OF 'EXPECTED' BUFFER
2169  46EF  EB               4.0        XCHG                  ;PUT POINTER IN REG 'D' & 'E'
2170  46F0  3A  DF  45      13.0        LDA   DCNTWD          ;GET # BYTES TO ASSEMBLE/4 SCLK CYCLE
2171  46F3  32  E2  45      13.0        STA   ABCNTR          ;SAVE FOR LATER TESTS
2172  46F6  E1              10.0        POP   H               ;RESET SAVED REGS
2173  46F7  C1              10.0        POP   B
2174  46F8  F1              10.0        POP   PSW
2175  46F9  C9              10.0        RET                   ;EXIT
2176                                  ;ROUTINE TO SHIFT CONTENTS OF ACCUMULATOR RIGHT OR LEFT EITHER 2 OR 4 TIMES
2177
2178  46FA  E6  FC           7.0  SHR4:   ANI   @374           ;STRIP OFF 2 BITS
2179  46FC  0F               4.0        RRC                   ;SHIFT RIGHT 2 TIMES
2180  46FD  0F               4.0        RRC
2181  46FE  E6  FC           7.0  SHR2:   ANI   @374           ;STRIP OFF 2 BITS
2182  4700  0F               4.0        RRC                   ;SHIFT RIGHT 2 TIMES
2183  4701  0F               4.0        RRC
2184  4702  C9              10.0        RET
2185
2186  4703  E6  3F           7.0  SHL4:   ANI   @77            ;SHIFT LEFT TWICE
2187  4705  07               4.0        RLC
2188  4706  07               4.0        RLC
2189  4707  E6  3F           7.0  SHL2:   ANI   @77
2190  4709  07               4.0        RLC                   ;SHIFT LEFT TWICE
2191  470A  07               4.0        RLC
2192  4708  C9              10.0        RET
2193
2194  470C                                  $
 (1)                                        ;*****************************************************************
```

```
2195                                    ;F11N -- ROUTINE TO FILL 'EXPECTED' BAL OUTPUT TABLE WITH 8 BYTES OF
2196                                    ;       11 NORMAL FORMAT DATA BASED ON DATA IN DDR STORAGE TABLE.  REG-
2197                                    ;       ISTERS D & E POINT TO STORAGE TABLE
2198  470C                              $
 (1)                                    ; *****************************************************************************
2199
2200  470C  06  08          7.0  F11N:  MVI     B,8             ;SETUP COUNTER
2201  470E  CD  48  47      18.0        CALL    FILTBL          ;FILL TABLE
2202  4711  C3  EC  46      10.0        JMP     GCDATX          ;EXIT
2203
2204  4714                              $
 (1)                                    ; *****************************************************************************
2205                                    ;F15N -- ROUTINE TO FILL 'EXPECTED' DATA TABLE WITH 15 NORMAL DATA BYTES
2206                                    ;       BASED ON DATA IN DDR STORAGE TABLE.
2207  4714                              $
 (1)                                    ; *****************************************************************************
2208
2209  4714  06  08          7.0  F15N:  MVI     B,8             ;SET COUNTER TO 8 BYTES
2210  4716  3A  F2  45      13.0        LDA     SDDRAC          ;GET DDR REG A DATA - BITS 7-0
2211  4719  32  F0  45      13.0        STA     SDDRBT          ;SAVE IN STORAGE DDR B - REVERSE A + B
2212  471C  3A  F3  45      13.0        LDA     SDDRBC          ;GET REAL DDR B
2213  471F  32  EF  45      13.0        STA     SDDRAT          ;SAVE IN DDR A STORAGE
2214  4722  CD  48  47      18.0        CALL    FILTBL          ;FILL COMPARE TALBE
2215  4725  3A  F2  45      13.0        LDA     SDDRAC          ;RESET DDR STORAGE FOR
2216  4728  32  EF  45      13.0        STA     SDDRAT          ;REGS A + B
2217  472B  3A  F3  45      13.0        LDA     SDDRBC
2218  472E  32  F0  45      13.0        STA     SDDRBT
2219  4731  C3  EC  46      10.0        JMP     GCDATX          ;EXIT
2220
2221  4734                              $
 (1)                                    ; *****************************************************************************
2222                                    ;FIMAG -- GENERATE COMPARE TABLE FOR BAL IMAGE MODE
2223  4734                              $
 (1)                                    ; *****************************************************************************
2224
2225  4734  06  04          7.0  FIMAG: MVI     B,4             ;SET BYTE COUNT TO
2226  4736  3A  F2  45      13.0        LDA     SDDRAC          ;GET DDR REG A DATA BYTE
2227  4739  32  F0  45      13.0        STA     SDDRBT          ;SAVE IN REG B STORAGE (ONLY NEG 8 BITS IN IMAGE)
2228  473C  CD  48  47      18.0        CALL    FILTBL          ;FILL COMPARE TABLE WITH EXPECTED IMAGE DATA
2229  473F  3A  F3  45      13.0        LDA     SDDRBC          ;RESET DDR REG 'B' DATA BYTE
2230  4742  32  F0  45      13.0        STA     SDDRBT
2231  4745  C3  EC  46      10.0        JMP     GCDATX          ;EXIT
2232
2233  4748                              $
 (1)                                    ; *****************************************************************************
2234                                    ;FILTBL -- ROUTINE COMMON TO 11/15 NORMAL, AND IMAGE MODE TO FILL THE
2235                                    ;       'EXPECTED' BAL OUTPUT BYTE DATA.  STORAGE BUFFER POINTED TO BY
2236                                    ;       D + E.
2237  4748                              $
 (1)                                    ; *****************************************************************************
2238
2239  4748  78                  4.0  FILTBL: MOV    A,B             ;PUT THE COUNT IN A
2240  4749  32  DF  45      13.0        STA     DCNTWD          ;SAVE IT
2241  474C  3A  EF  45      13.0  FILTBX: LDA     SDDRAT          ;GET DDR REG 'A' - BITS 7-0 DATA
```

```
2242  474F  12                7.0          STAX    D               ;SAVE IN TABLE
2243  4750  13                6.0          INX     D               ;POINT TO NEXT TABLE LOCATION
2244  4751  05                4.0          DCR     B               ;DECREMENT BYTE COUNT
2245  4752  3A  F0  45        13.0          LDA     SDDRBT          ;GET DDR REG 'B' - BITS 15-8 DATA
2246  4755  12                7.0          STAX    D               ;SAVE IN TABLE
2247  4756  13                6.0          INX     D               ;POINT TO NEXT TABLE LOCATION
2248  4757  05                4.0          DCR     B               ;DECREMENT BYTE COUNT
2249  4758  C2  4C  47       10.0          JNZ     FILTBX          ;DONE? JUMP BACK IF MORE TO DO.
2250  475B  AF                4.0          XRA     A
2251  475C  32  E6  45        13.0          STA     LFTTST          ;CLEAR CHECK FLAG FOR 'WMC LEFT' TEST
2252  475F  32  E7  45        13.0          STA     LFTFLG
2253  4762  C9                10.0          RET                     ;YES - EXIT.
2254  4763                                 S
 (1)                                       ;*****************************************************************************
2255                                       ;LDBCNT -- ROUTINE TO LOAD THE BYTE COUNTER WITH NUMBER OF EXPECTED
2256                                       ;       BYTES BASED ON 60 SCLKS OF DATA AND THE SELECTED FORMAT CODE.
2257  4763                               ;
 (1)                                       ;*****************************************************************************
2258
2259  4763  0E  10                7.0  LDBCNT: MVI     C,DTEND-CDATBL/3        ;GET AMOUNT OF DATA PATTERNS
2260  4765  47                4.0          MOV     B,A             ;PUT THE COUNT IN REG B
2261  4766  AF                4.0          XRA     A               ;CLEAR ACCUMULATOR
2262  4767  80                4.0  LDBC1:  ADD     B               ;ADD IN BYTE COUNT (MULTIPLY)
2263  4768  0D                4.0          DCR     C               ;DOWNCOUNT
2264  4769  C2  67  47       10.0          JNZ     LDBC1           ;JUMP IF LOOP NOT DONE
2265  476C  32  E0  45        13.0          STA     IMBYTC          ;AND SAVE IN IMAGE STORAGE FRO BITS 7-0
2266  476F  3C                4.0          INR     A               ;FIX BYTE COUNTER FOR INTEL 8253 CHIP
2267  4770  D3  D4            10.0          OUT     BYTCNT          ;SAVE IN BYTE COUNTER
2268  4772  AF                4.0          XRA     A               ;CLEAR BITS 15-8 OF COUNTER
2269  4773  D3  D4            10.0          OUT     BYTCNT          ;LOAD BITS 15-8
2270  4775  C9                10.0          RET                     ;EXIT
2271
2272  4776                                 S
 (1)                                       ;*****************************************************************************
2273                                       ;F10D -- ROUTINE TO CALCULATE THE 10 BYTES EXPECTED FROM BAL WHEN IN 10
2274                                       ;       DUMP MODE AND HAVE 4 SCLKS (2 36 BIT WORDS OR 4 18 BIT WORDS).
2275  4776                               ;
 (1)                                       ;*****************************************************************************
2276
2277  4776  3E  0A              7.0  F10D:   MVI     A,10            ;FILL 10 BYTES OF DATA IN TABLE
2278  4778  32  DF  45        13.0          STA     DCNTWD          ;SAVE THE COUNT
2279  477B  CD  95  47        18.0  F10DA:  CALL    BYTATD          ;GET BYTES 'A' TO 'D'
2280  477E  CD  EF  47        18.0          CALL    F10DE           ;GET BYTE 'E'
2281  4781  C2  7B  47        10.0          JNZ     F10DA           ;LOOP BACK IF MORE TO DO
2282  4784  C3  EC  46        10.0          JMP     GCDATX          ;THEN EXIT
2283
2284  4787                                 S
 (1)                                       ;*****************************************************************************
2285                                       ;F10C -- ROUTINE TO CALCULATE 10 COMPAT. MODE BYTES 'A' TO 'D'
2286  4787                               ;
 (1)                                       ;*****************************************************************************
2287
2288  4787  3E  08              7.0  F10C:   MVI     A,8
2289  4789  32  DF  45        13.0          STA     DCNTWD          ;SAVE THE BYTE COUNT/4 SCLK
```

```
2290  478C  CD  95  47   18.0  F10CA:  CALL   BYTATD          ;GET BYTES 'A' TO 'D'
2291  478F  C2  8C  47   10.0          JNZ    F10CA
2292  4792  C3  EC  46   10.0          JMP    GCDATX          ;THEN EXIT
2293                               ;CALC. 10 MODE BYTE 'A' - BITS 0 TO 7
2294
2295  4795  47                4.0  BYTATD: MOV    B,A             ;COPY THE COUNT
2296  4796  3A  F0  45       13.0          LDA    SDDRBT          ;GET DDR REG 'B' DATA
2297  4799  CD  FE  46       18.0          CALL   SHR2            ;SHIFT ACCUMULATOR RIGHT 2 PLACES
2298  479C  4F                4.0          MOV    C,A             ;SAVE RESULT
2299  479D  3A  F1  45       13.0          LDA    SDDRCT          ;GET BIT 17 + 16 DATA BITS
2300  47A0  E6  03            7.0          ANI    3
2301  47A2  0F                4.0          RRC
2302  47A3  0F                4.0          RRC                    ;PUT 16 + 17 DATA IN BIT 6 + 7 POSITION
2303  47A4  B1                4.0          ORA    C               ;COMBINE FOR BYTE 'A'
2304  47A5  12                7.0          STAX   D               ;SAVE IN TABLE
2305  47A6  13                6.0          INX    D               ;POINT TO NEXT ENTRY IN TABLE
2306  47A7  05                4.0          DCR    B               ;DECREMENT BYTE COUNT
2307
2308                               ;CALC. 10 MODE BYTE 'B' - BITS 8 TO 15
2309
2310  47A8  3A  F0  45       13.0  F10MB:  LDA    SDDRBT
2311  47AB  E6  03            7.0          ANI    3               ;SAVE BYTS ''17'' + ''16''
2312  47AD  0F                4.0          RRC
2313  47AE  0F                4.0          RRC                    ;POSITION
2314  47AF  4F                4.0          MOV    C,A             ;SAVE FOR LATER
2315  47B0  3A  EF  45       13.0          LDA    SDDRAT          ;GET DDR REG 'A' DATA
2316  47B3  CD  FE  46       18.0          CALL   SHR2            ;SHIFT RIGHT 2 PLACES
2317  47B6  B1                4.0          ORA    C               ;COMBINE TO MAKE BYTE 'B'
2318  47B7  12                7.0          STAX   D               ;SAVE BYTE 'B' IN TABLE
2319  47B8  13                6.0          INX    D               ;UPDATE TABLE POINTER
2320  47B9  05                4.0          DCR    B               ;DECREMENT BYTE COUNT
2321
2322                               ;CALC. 10 MODE BYTE 'C' - BITS 16 TO 23
2323
2324  47BA  3A  EF  45       13.0  F10MC:  LDA    SDDRAT          ;GET DDR REG 'A' BYTE
2325  47BD  E6  03            7.0          ANI    3               ;SAVE BITS ''16'' + ''17''
2326  47BF  0F                4.0          RRC
2327  47C0  0F                4.0          RRC
2328  47C1  4F                4.0          MOV    C,A             ;SAVE FOR LATER
2329  47C2  3A  F1  45       13.0          LDA    SDDRCT          ;GET DDR REG 'C' BYTE
2330  47C5  CD  03  47       18.0          CALL   SHL4            ;SHIFT LEFT 4 PLACES TO 5 + 4
2331  47C8  B1                4.0          ORA    C               ,COMBINE FOR BITS 16-19
2332  47C9  4F                4.0          MOV    C,A             ;SAVE FOR LATER
```

```
2334  47CA  3A  F0  45   13.0          LDA   SDDRBT          ;GET DDR REG 'B' BYTE
2335  47CD  CD  FA  46   18.0          CALL  SHR4            ;RIGHT JUSTIFY 4 PLACES
2336                                                        ;FOR BITS "20" - "23"
2337  47D0  B1                 4.0      ORA   C               ;COMBINE TO MAKE BYTE 'C'
2338  47D1  12                 7.0      STAX  D               ;SAVE 10 DUMP BYTE 'C' IN TABLE
2339  47D2  13                 6.0      INX   D               ;POINT TO NEXT TABLE ENTRY
2340  47D3  05                 4.0      DCR   B               ;DECREMENT BYTE COUNT
2341                                 ;CALC. 10 MODE BYTE 'D' - BITS 24 TO 31
2342
2343  47D4  3A  F0  45   13.0  F10MD:  LDA   SDDRBT          ;GET DDR REG 'B' DATA
2344  47D7  CD  03  47   18.0          CALL  SHL4            ;LEFT JUSTIFY FOR BITS 24 - 27
2345  47DA  4F                 4.0      MOV   C,A             ;SAVE FOR LATER
2346  47DB  3A  EF  45   13.0          LDA   SDDRAT          ;GET DDR REG 'A' DATA
2347  47DE  CD  FA  46   18.0          CALL  SHR4            ;RIGHT JUSTIFY REMAINING
2348                                                        ;4 HIGH BITS TO BIT 0 FOR
2349                                                        ;BITS "28" TO "31"
2350  47E1  B1                 4.0      ORA   C               ;COMBINE TO MAKE BYTE 'D'
2351  47E2  12                 7.0      STAX  D               ;SAVE IN TABLE
2352  47E3  13                 6.0      INX   D               ;POINT TO NEXT TABLE ENTRY
2353  47E4  3E  FF             7.0      MVI   A,@377
```

```
2355  47E6  32  E6  45   13.0          STA     LFTTST          ;SET TO TEST 'LEFT'
2356  47E9  32  E7  45   13.0          STA     LFTFLG          ;'WMC LEFT' SET
2357  47EC  05               4.0      DCR     B               ;DECREMENT COUNT OF BYTES/SCLK
2358  47ED  78               4.0      MOV     A,B             ;AND PUT COUNT IN REG A
2359  47EE  C9              10.0      RET
2360
2361                                 ;CALC. REMAINING 10 DUMP BYTE 'E' - BITS 32 TO 35
2362
2363  47EF  3A  EF  45   13.0  F10DE:  LDA     SDDRAT          ;GET DDR REG 'A' DATA
2364  47F2  E6  0F           7.0      ANI     $F              ;SAVE 4 BITS
2365  47F4  12               7.0      STAX    D               ;SAVE 10 DUMP BYTE 'E'
2366  47F5  13               6.0      INX     D               ;POINT TO NEXT TABLE ENTRY
2367  47F6  05               4.0      DCR     B               ;DECREMENT BYTE COUNT
2368  47F7  78               4.0      MOV     A,B             ;COPY TO REG A
2369  47F8  C9              10.0      RET
2370  47F9                         S
  (1)                                ;*****************************************************************************
2371                                 ;F10HD -- ROUTINE TO CALCULATE THE 'EXPECTED' DATA BYTE OUTPUT FROM BAL
2372                                 ;       WHEN IN 10 HIGH DENSITY MODE.  4 SCLKS GENERATE 9 BYTES OF OUT-
2373                                 ;       PUT.  D & E POINT TO STORAGE TABLE.
2374  47F9                         S
  (1)                                ;*****************************************************************************
2375
```

```
2377                                        ;CALC. 10 HD BYTE 'A' - BITS 0 TO 7
2378
2379  47F9  3E  09        7.0   F10HD:  MVI   A,9           ;GET BYTE COUNT/4 SCLKS
2380  47FB  32  DF  45    13.0          STA   DCNTWD        ;SAVE IT
2381  47FE  CD  95  47    18.0          CALL  BYTATD        ;GET BYTES 'A' TO 'D'
2382
2383                                        ;CALC. 10 HD BYTE 'E' - BITS 32 TO BITS 3 OF NEXT WORD
2384
2385  4801  3A  EF  45    13.0  F10HDE: LDA   SDDRAT        ;GET DDR 'A'
2386  4804  CD  03  47    18.0          CALL  SHL4
2387  4807  4F             4.0          MOV   C,A           ;SAVE
2388  4808  3A  F1  45    13.0          LDA   SDDRCT        ;GET DDR 'C'
2389  480B  CD  07  47    18.0          CALL  SHL2
2390  480E  B1             4.0          ORA   C             ;COMBINE FOR BITS 7 TO 2
2391  480F  4F             4.0          MOV   C,A           ;SAVE
2392  4810  3A  F0  45    13.0          LDA   SDDRBT        ;GET DDR 'B'
2393  4813  E6  C0         7.0          ANI   @300          ;SAVE 2 HIGH BITS
2394  4815  07             4.0          RLC                 ;POSITION TO BITS
2395  4816  07             4.0          RLC                 ;1 + 0
2396  4817  B1             4.0          ORA   C             ;COMBINE TO MAKE BYTE 'E'
2397  4818  12             7.0          STAX  D             ;SAVE 10 HD BYTE 'E' IN TABLE
```

```
2399  4819  13              6.0          INX     D               ;POINT TO NEXT TABLE ENTRY
2400
2401                                   ;CALC. 10 HD BYTE 'F' - BITS 4 TO 11 OR WORD 2
2402
2403  481A  3A  F0  45       13.0  F10HDF: LDA     SDDRBT          ;GET DDR 'B'
2404  481D  CD  07  47        18.0         CALL    SHL2
2405  4820  4F               4.0          MOV     C,A             ;SAVE
2406  4821  3A  EF  45       13.0         LDA     SDDRAT          ;GET DDR 'A'
2407  4824  E6  C0           7.0          ANI     @300            ;SAVE 2 BITS
2408  4826  07               4.0          RLC                     ;POSITION IN THE LEFT
2409  4827  07               4.0          RLC
```

```
2411  4828  B1                    4.0          ORA     C               ;COMBINE TO MAKE BYTE 'F'
2412  4829  12                    7.0          STAX    D               ;SAVE 10 HD BYTE 'F'
2413  482A  13                    6.0          INX     D               ;POINT TO NEXT TABLE ENTRY
2414
2415                                     ;CALC. 10 HD BYTE 'G' - BITS 12 TO 19 OF 2ND WORD
2416
2417  482B  3A  EF  45            13.0  F10HDG: LDA     SDDRAT          ;GET DDR 'A'
2418  482E  CD  07  47            18.0         CALL    SHL2
2419  4831  4F                    4.0          MOV     C,A             ;SAVE
2420  4832  3A  F1  45            13.0         LDA     SDDRCT          ;GET DDR 'C'
2421  4835  E6  03                 7.0          ANI     3               ;SAVE 2 BITS
2422  4837  B1                    4.0          ORA     C               ;COMBINE TO MAKE BYTE 'G'
2423  4838  12                    7.0          STAX    D               ;SAVE IN TABLE
2424  4839  13                    6.0          INX     D               ;POINT TO NEXT TABLE ENTRY
2425                                     ;CALC. 10 HD BYTE 'H' - BITS 20 - 27
2426
2427  483A  3A  F0  45            13.0  F10HDH: LDA     SDDRBT          ;GET DDR 'B'
2428  483D  12                    7.0          STAX    D               ;STORE IN TABLE AS BYTE 'H'
2429  483E  13                    6.0          INX     D               ;POINT TO NEXT BYTE ENTRY
2430
2431                                     ;GET + SAVE LAST BYTE OF 10 HD MODE - BITS 28 - 35 OF 2ND WORD
2432
2433  483F  3A  EF  45            13.0  F10HDI: LDA     SDDRAT          ;GET DDR 'A'
2434  4842  12                    7.0          STAX    D               ;STORE IN TABLE
2435  4843  C3  EC  46            10.0         JMP     GCDATX          ;GO BACK TO COMMON EXIT
2436
2437  4846                                    $
 (1)                                 ; *****************************************************************************
2438                                     ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
2439                                     ;       'CLK' BIT TO A '1' THEN A '0'.
2440  4846                               ;
 (1)                                 ; *****************************************************************************
2441
2442  4846  3A  DE  45            13.0  CLKSYS: LDA     CCTLWD          ;GET SOFTWARE CLOCK CONTROL IMAGE
2443  4849  F6  40                 7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
2444  484B  D3  F0                10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
2445  484D  E6  3F                 7.0          ANI     @77             ;STRIP OFF CLOCK BIT
2446  484F  D3  F0                10.0     .    OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
2447  4851  C9                    10.0          RET                     ;EXIT - CLOCK CYCLE COMPLETE
2448
2449        0000                               .END
```

```
A      =%0007   ABCNT   45E1     ABCNTR  45E2     ADATA  = 0094
AMTIEP= 0001    AMTIE7= 0002     ARAIDF= 0098     ASAVE   4F9B
ATTCD   4F97    AXNUM   4F91     B      =%0000    BADST  = 0090
BALDAT  46BB    BIT0   = 0001    BIT1   = 0002    BIT15  = 8000
BIT2   = 0004   BIT3   = 0008    BIT4   = 0010    BIT5   = 0020
BIT6   = 0040   BIT7   = 0080    BIT8   = 0100    BIT9   = 0200
BRKPBF= 4F0A    BRKRAM= 4F10     BRKSTR= 4E60     BRKXCT= 4F00
BSAVE   4F9C    BYTATD  4795     BYTCNT= 00D4     BYTEH   4F24
BYTEL   4F23    C      =%0001    CASCT   4F21     CASCTL= 00A0
CASSTA= 00A0    CATTH  = 0089    CATTL  = 0088    CBUSST= 00A1
CBYTH  = 0088   CBYTL  = 008A    CCTLWD  45DE     CDATBL  4688
CDG1H  = 0087   CDG1L  = 0086    CDG2H  = 0093    CDG2L  = 0092
CDG3H  = 0095   CDG3L  = 0094    CDVTH  = 008D    CDVTL  = 008C
CHPTIE= 0028    CH0TIE= 0020     CH1TIE= 0021     CH2TIE= 0022
CH3TIE= 0023    CH4TIE= 0024     CH5TIE= 0025     CH6TIE= 0026
CH7TIE= 0027    CKLOP  = 2815    CKSCLK  441A     CKSC1   4429
CLKCNT  45EA    CLKCTL= 00F0     CLKON   45E8     CLKSYS  4846
CLOCK   4F26    CMC0H  = 0099    CMC0L  = 0098    CMC1H  = 009B
CMC1L  = 009A   CMC2H  = 009D    CMC2L  = 009C    CMC3H  = 009F
CMC3L  = 009E   CMINH  = 0097    CMINL  = 0096    CNTCTL= 00D7
CRCWRD= 0018    CSAVE   4F9D     CSRLH  = 0091    CSRLL  = 0090
CTCH   = 0085   CTCL   = 0084    CTSTH  = 008F    CTSTL  = 008E
CXCTH  = 0081   CXCTL  = 0080    CXINH  = 0083    CXINL  = 0082
C.     = 0001   C.AVAI= 0080     C.DP   = 0008    C.DSE  = 0010
C.DTU  = 0003   C.DVA  = 0008    C.FAIL= 00FC     C.FMT  = 0070
C.FNCT= C03E    C.GO   = 0001    C.INTC= 00FE     C.MAIN= 0020
C.NSA  = 0080   C.RCT  = 00FC    C.SER  = 0080    C.SHR  = 0040
C.SKPC= 000F    C.TAPE= 0040     C.WCS  = 0002    D      =%0002
DATACT= 00D0    DATPE   4502     DATPEC  450D  G  DATPEE  4513
DATPO   44EC    DATPOC  44FA  G  DATPOE  4500    DBUS    4F28
DBUSCT= 00C0    DBUSST= 00C0     DCNTWD  45DF     DDRA   = 00D8
DDRAIN= 0010    DDRB   = 00D9    DDRBIN= 0002     DDRC   = 00DA
DDRCIN= 0001    DDRCO  = 0088    DDRCTL= 00DB     DELOOP  456C
DELP1   4579    DIAFLG  4F22     DIAGPG= 4300     DIAGRM= 4F90
DIARD  = 000B   DONE1  = 0045    DONINT= 0010     DPNUM   45E5
DSAVE   4F9E    DSE    = 0006    DTEND   46B8     D.ATH0= 0001
D.ATH1= 0002    D.EOTD= 0010     D.LAGC= 0020     D.NOTW= 0040
D.NTHR= 0004    D.TACH= 0008     D.WR4  = 0080    E      =%0003
EABCNT  45E3    ECCBAD= 0042     ECCCHK  4587     ECCCNT  45E9
ECCCOR= 0019    ECCOK  = 0041    ECCON   459B  G  ECCON1  45A1
ECCSTA= 001A    ECCTST= 000E     EDATA  = 0095    EOTCLR= 0003
ERFLG   4F93    ERLP   = 2809    ERLPA  = 280F    ERLPB  = 2812
ERLPE  = 280C   ERNUM   4F90     ERRCNT= 00D6     ESAVE   4F9F
E.ACRC= 0010    E.AMT  = 0020    E.CDP  = 0080    E.CRC  = 0080
E.PNTR= 0008    E.RPE  = 0040    E.STEC= 0001     E.TTEC= 0002
E.UNC  = 0004   FIFORD= 006A     FILTBL  4748     FILTBX  474C
FIMAG   4734    FMTTBL  46C5     FORMAT  4F25     FORMT   45E4
FWDTST= 0061    F10C    4787     F10CA   478C     F10D    4776
F10DA   477B    F10DE   47EF     F10HD   47F9     F10HDE  4801
F10HDF  481A    F10HDG  482B     F10HDH  483A     F10HDI  483F
F10MB   47A8    F10MC   47BA     F10MD   47D4     F11N    470C
F15N    4714    GCDAT   46D3     GCDATX  46EC     GCRID  = 0089
GCRSET= 0002    GENDAT  45F8     GENOUT  4623     GOODTM= 0092
H      =%0004   HLSAVE  4FA0     IE     = 0008    IMBYTC  45E0
```

```
INTSTA= 00E0        ITERA   4F9A        I.PWR  = 0020        I.RMPE= 0040
I5.5  = 0010        I6.5  = 0020        I7.5   = 0040        KCALL = 005F
KCLR  = 007B        KDEP  = 003F        KENAB  = 0078        KEXAM = 003E
KEYBRD= 00C8        KEY1  = 0078        KEY10  = 006D        KEY11 = 006E
KEY12 = 006F        KEY13 = 005C        KEY14  = 005D        KEY15 = 005E
KEY16 = 005F        KEY17 = 003C        KEY18  = 003D        KEY19 = 003E
KEY2  = 0079        KEY20 = 003F        KEY3   = 007A        KEY4  = 007B
KEY5  = 0074        KEY6  = 0075        KEY7   = 0076        KEY8  = 0077
KEY9  = 006C        KINTA = 006F        KLDAD  = 003D        KN0   = 003C
KN1   = 005C        KN2   = 005D        KN3    = 005E        KN4   = 006C
KN5   = 006D        KN6   = 006E        KN7    = 0074        KN8   = 0075
KN9   = 0076        KU2   = 0079        KU3    = 007A        KU8   = 0077
L     =%0005        LBLANK= 000F        LCE    = 000B        LCH   = 000C
LCL   = 000D        LCP   = 000E        LC0    = 0000        LC1   = 0001
LC2   = 0002        LC3   = 0003        LC4    = 0004        LC5   = 0005
LC6   = 0006        LC7   = 0007        LC8    = 0008        LC9   = 0009
LDBCNT  4763        LDBC1   4767        LDLEDA= 00CA        LDLEDB= 00CB
LDLEDC= 00CC        LDLEDD= 00CD        LDLEDE= 00CE        LDLEDF= 00CF
LFTCHK  45A4        LFTCMP  45C3        LFTFLG  45E7        LFTOFF  45CB
LFTOK1  45D7  G     LFTOK2  45BC  G     LFTON   45B0        LFTTST  45E6
LKDIAG= 2800        LKKBD = 004C        LKKEY  = 0049        LKLWMG= 0058
LKLWMP= 0055        LKLWPG= 0052        LKLWPP= 004F         LKMOD7= 0046
LKOPR = 0046        LPFLG   4F94        LPNUM   4F92         M     =%0006
MBSEL = 00E0        MB.A  = 0008        MB.B   = 0004        MEMTOP= 4FFF
MINUS = 000A        MM    = 8000        MSE    = 0008        MSGN  = 000D
MTACLR= 0000        MT.ARA= 0020        MT.CPE= 0080         MT.DSE= 0001
MT.FWD= 0040        MT.INH= 0008        MT.LWR= 0004         MT.MOT= 0002
MT.NWT= 0080        MT.PEC= 0040        MT.PSB= 0004         MT.PS0= 0001
MT.PS1= 0002        MT.REV= 0020        MT.WRT= 0010         MT.Z  = 0008
M.ATA = 0080        M.CAPE= 0020        M.CONT= 0080         M.DEM = 0020
M.EBL = 0010        M.EXC = 0008        M.FAIL= 0008         M.ILR = 0010
M.INIT= 0010        M.OCC = 0020        M.ONLI= 0001         M.PE  = 0040
M.PORT= 0080        M.RDEN= 0002        M.RDPE= 0008         M.RUN = 0004
M.SCLK= 0001        M.TRA = 0040        M.UNIT= 0007         M.WCLK= 0040
M.WCLN= 0080        M.WREN= 0080        M5.5  = 0001         M6.5  = 0002
M7.5  = 0004        NOTCAP= 0088        OKAY  = 00FF         OPRRAM= 4300
OPSTRT= 0058        OPVER = 0040        PADCNT= 00D5         PADCRC= 0080
PDIAG = 0048        PEID  = 008A        PENAB = 004C         PESET = 0001
PL    = 00B1        PRDD  = 004C        PRENF = 009C         PS    = 00B2
PSTAT = 0048        PSW   =%0009        P.AMTP= 0001         P.BCTC= 0040
P.CMDP= 0020        P.INTE= 0080        P.LCS = 0040         P.LWR = 0020
P.RDP = 0002        P.RPEN= 0004        P.RPST= 0002         P.RPOE= 0020
P.RP1E= 0010        P.RP2E= 0020        P.RP3E= 0010         P.SING= 0080
P.STAT= 0002        P.STPE= 0080        P.TACH= 0008         P.TUPR= 0010
P.WCSP= 0004        P.WDS = 0040        P.WFLP= 0001         P.WPEN= 0010
P.WPOE= 0008        P.WP1E= 0004        P.WP2E= 0008         P.WP3E= 0004
P.5VOK= 0002        QUE   = 281B        QUEM  = 281E         RAMT  = 0010
RARA  = 0006        RARAI = 0004        RCHBD0= 0048         RCHBD1= 0047
RCHOK = 0046        RC4TST= 000C        RCLRT = 000D         RCMD  = 0008
RCMLP = 0003        RCONT = 0080        RDATA = 0017         RDCLK = 0010
RDON  = 0011        READG = 0007        REND  = 0014         REQST = 2806
RESCHR= 00D1        REVTST= 0064        REWIND= 0004         RFIFOL= 0008
RGCLK = 0002        RGCRI = 0003        RIBG  = 0001         RILL  = 0012
RINST = 000C        RMCTST= 0008        RMK2  = 0013         RNOP  = 0000
```

```
RPATH = 0001        RPBAD = 0044        RPCHI = 0001        RPCLK = 0003
RPCTL = 0009        RPEI  = 0002        RPFAIL= 0000        RPF1  = 009D
RPF2  = 009E        RPOK  = 0043        RPOSTN= 0016        RPSTA = 0015
RRCMT = 000A        RSTAT = 0002        RTIEB = 000A        RTIER = 0030
RTM   = 0005        RUNKI = 0009        RUPTST= 005E        RWDUNL= 0005
R.AMT = 0001        R.BOP = 0008        R.DATA= 0040        R.DON = 0002
R.DRDY= 0010        R.END = 0010        R.ILL = 0004        R.JVOK= 0004
R.MK2 = 0008        R.PLOD= 0008        R.PL00= 0010        R.PL01= 0020
R.POST= 0020        R.STNM= 0002        R.STOP= 0004        R.STPC= 0001
R.TBJN= 0080        R.TSTD= C040        R.VOK = 0080        R00H  = 0081
R00L  = 0080        R01H  = 0083        R01L  = 0082        R02H  = 0085
R02L  = 0084        R03H  = 0087        R03L  = 0086        R04H  = 0089
R04L  = 0088        R05H  = 008B        R05L  = 008A        R06H  = 008D
R06L  = 008C        R07H  = 008F        R07L  = 008E        R10H  = 0091
R10L  = 0090        R11H  = 0093        R11L  = 0092        R12H  = 0095
R12L  = 0094        R13H  = 0097        R13L  = 0096        R14H  = 0099
R14L  = 0098        R15H  = 009B        R15L  = 009A        R16H  = 009D
R16L  = 009C        R17H  = 009F        R17L  = 009E        R7.5  = 0010
SCCNTR  45EB        SCKB1   4402        SCKGON  43EF        SCKWT   43C9
SCKWTC  43E6  G     SCKWTE  4419        SCKWT1  43D6        SCLKBI  43BE
SDDCOM  4678        SDDRA   45EC        SDDRAC  45F2        SDDRAT  45EF
SDDRB   45ED        SDDRBC  45F3        SDDRBT  45F0        SDDRC   45EE
SDDRCC  45F4        SDDRCT  45F1        SELCLR= 0000        SETATA= 00A1
SETDDR  4627        SHL2    4707        SHL4    4703        SHR2    46FE
SHR4    46FA        SID   = 0080        SINTST  45F7        SLFTDD  4666
SOD   = 0080        SOE   = 0040        SP    =%0008        SSCLK = 0040
SSTEP = 0005        STACK = 4FFF        STATRM= 4F20        STPCT   4F20
STRSP = 5000        SWMCST  45F5        SXFER   45F6        TADR00= 0080
TADR01= 0081        TADR02= 0082        TADR03- 0083        TADR04= 0084
TADR05= 0085        TADR06= 0086        TADR07= 0087        TADR10= 0088
TADR11= 0089        TADR12- 008A        TADR13= 008B        TAMT  = 0044
TASEL = 0080        TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008
TC.LWR= 0004        TC.REV= 0020        TC.WRT= 0010        TEMP    4F99
TMF   - 0099        TMRDY = 0040        TRKENA= 00D2        TSET  = 2803
TSTEND= 2818        TSTS  = 0040        TST1    4300        TST1A   4371
TST1B   437C        TST1BC  43A3  G     TST1C   439A        TST1D   43AC
TST1DC  43B5  G     TST1L   430C        TST1ST  4305        TST1XC  44D4  G
TUSEL0= 00D1        TUSEL1= 00D2        TU78  = 0010        T.ATH0= 0001
T.ATH1= 0002        T.BOT = 0004        T.EOT = 0002        T.FPT = 0001
T.NTHR= 0004        T.ONL = 0020        T.PES = 0008        T.PSBJ= 0020
T.PSOJ= 0008        T.PS1J= 0010        T.RDY = 0080        T.RDYO= 0040
T.RWD = 0010        T.SCLK= 0002        T1RST   4364        UIBG  = 00A1
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT- 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN- 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT - 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
W.SKIP- 000F        W.WRIT= 0008        W.XFER= 0020        X     =%000A
XFRBIT  442D        XFRB1   4435        XFRB2   4453        XFRB2C  4469  G
XFRB3   4472        XFRB3A  4479        XFRB3C  448F  G     XFRB4   4498
XFRB4P  44DA        XFRB5   4514        XFRB5C  4543  G     XFRGON  444B  G
X.DONN= 0080        X.ENAB  0040        X.PEPE= 0002        X.ROME= 0001
X.WCLK= 0001        Y     %000B         .     = 4852
```

ERRORS DETECTED: 0

*WMC3.A78/PTP,WMC3=NLIST,PARAM,MACRO,LIST,WMC3
RUN-TIME: 4 6 0 SECONDS
CORE USED: 10K

.MAIN. ROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:07
PARAM.M8' 'ABLE OF CONTENTS SEQ 0136

```
                                .TITLE  WMC4 - WMC ECC, ACRC, CRC, AND RESID CHAR TESTS
                                .SBTTL  TEST 1 - WRITE MICRO CHARACTER GENERATION TEST IN GCR MODE
                                ;ID     WMC4-WRITE MICRO CONTROLLER PART #4
4300                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR CHARACTER TESTS
4300                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST IS DESIGNED TO TEST THE WRITE MICROCONTROLLER CHARACTER GENERATION
                                ;*LOGIC.  A KNOWN DATA PATTERN IS FED INTO THE BYTE ASSEMBLY LOGIC FROM
                                ;*THE DIAGNOSTIC DATA REGISTERS - 'DDR' - TO PROVIDE AN 8 BIT WIDE
                                ;*REPLACEMENT FOR MASSBUS DATA IN IMAGE MODE.
                                ;*
                                ;*THE BYTE ASSEMBLY LOGIC WILL 'BREAK UP' THE INPUT DATA (8 BITS)
                                ;*INTO THE DESIRED BYTE FORMAT TO BE WRITTEN ON TAPE.
                                ;*
                                ;*THIS TEST WILL CHECK THE OUTPUT OF THE WRITE MICRO TO BE CORRECT
                                ;*IN REGARD TO THE ECC CHARACTERS GENERATED AND THE PROPER LOCATION
                                ;*OF THESE CHARACTERS IN THE DATA STREAM.
4300                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   INIT THE TEST
                                ;*   SET FORMAT INDEX NUMBER FOR IMAGE MODE
                                ;*   BGNDO
                                ;*   :   INIT DATA PATTERN NUMBER FOR PASS
                                ;*   :   RESTART WMC & SET CLOCK CONTROL TO SINGLE STEP
                                ;*   :   INIT WMC BYTE COUNTER + ERROR CODE COUNTER AND PAD COUNTER CONTROL
                                ;*   :   CLEAR ERROR CODE COUNTER
                                ;*   :   CLEAR PROGRAM FLAGS AND STORAGE AREAS
                                ;*   :   SET DDR CONTROL TO 'OUT' SO DATA INJECTION POSSIBLE
                                ;*   :   GENERATE DATA TO DDR REGISTERS AND CHECK STORAGE
                                ;*   :   INIT SIMULATED 'ECC' GENERATED CHARACTER
                                ;*   :   INIT SIMULATED 'CRC' GENERATED CHARACTER
                                ;*   :   INIT SIMULATED 'ACRC' GENERATED CHARACTER
                                ;*   :   CLOCK WMC THROUGH ITS RESTART AREA-WMC ROM PROGRAM
                                ;*   :   CALL SUBROUTINE LKMOD7-CALCULATE 'MOD7' + 'MOD32'
                                ;*   :   LOAD RESCHR WITH MOD7 REMAINDER + MOD32
                                ;*   :   CALCULATE PROEPR NUMBER OF PAD CHARACTERS AND LOAD PAD COUNTER
                                ;*   :   LOAD DATACTL WITH FORMAT 5 (IMAGE) AND SKIP COUNT 0
                                ;*   :   SET WMCCTL TO 'WRITE' & 'ENABLE'-START WMC ROM PGM
                                ;*   :   BGNDO
                                ;*   :   :   CLOCK SYSTEM
                                ;*   :   :   IF WMCSTA HAS NO 'XFER' BIT SET
                                ;*   :   :   :   THEN-UPDATE TIMEOUT COUNTERS
                                ;*   :   :   :   :   IF TIMEOUT DETECTED
                                ;*   :   :   :   :   :   THEN-REPORT LACK OF ACTIVITY IN SPECIFIED TIME
                                ;*   :   :   :   :   :   ELSE-CONTINUE
```

```
1374                                    ;*  :  :  :  :  ENDIF
1375                                    ;*  :  :  :  ELSE-GET WMCSTA WORD + SAVE FOR LATER CHECKING
1376                                    ;*  :  :  :  :  BGNDO
1377                                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 1ST NIBBLE OF DATA
1378                                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1379                                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' BIT
1380                                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1381                                    ;*  :  :  :  :  :  ENDIF
1382                                    ;*  :  :  :  :  :  DO UNTIL 'XFER' CLEARS OR TIMEOUT
1383                                    ;*  :  :  :  :  ENDDO
1384                                    ;*  :  :  :  :  BGNDO
1385                                    ;*  :  :  :  :  :  CLOCK SYSTEM TO BEGIN 2ND NIBBLE OF DATA
1386                                    ;*  :  :  :  :  :  IF > 128 CLOCKS
1387                                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO SET 'XFER' FOR 2ND NIBBLE
1388                                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1389                                    ;*  :  :  :  :  :  ENDIF
1390                                    ;*  :  :  :  :  :  DO UNTIL 'XFER' SETS OR TIMEOUT
1391                                    ;*  :  :  :  :  ENDDO
1392                                    ;*  :  :  :  :  GET PARITY BIT FROM INTSTA WORD FOR ASSEMBLED BYTE
1393                                    ;*  :  :  :  :  BGNDO
1394                                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 2ND NIBBLE
1395                                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1396                                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' ON 2ND NIBBLE
1397                                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1398                                    ;*  :  :  :  :  :  ENDIF
1399                                    ;*  :  :  :  :  :  DO UNTIL 'XFER' IS CLEAR OR TIMEOUT
1400                                    ;*  :  :  :  :  ENDDO
1401                                    ;*  :  :  :  :  CLOCK SYSTEM TWICE MORE TO 'SET' INTERNAL STATUS
1402                                    ;*  :  :  :  :  IF WRTDAT BYTE IS AN 'ECC' CHARACTER
1403                                    ;*  :  :  :  :  :  THEN-GET ECC CHARACTER FROM WRTDAT
1404                                    ;*  :  :  :  :  :  :  IF ECC BYTE=SIMULATED BYTE
1405                                    ;*  :  :  :  :  :  :  :  THEN-GET ECC BYTE PARITY BIT
1406                                    ;*  :  :  :  :  :  :  :  :  IF PARITY ON ECC IS OK
1407                                    ;*  :  :  :  :  :  :  :  :  :  THEN-EXIT DO LOOP
1408                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON ECC BYTE
1409                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1410                                    ;*  :  :  :  :  :  :  :  ELSE-REPORT 'ECC' GENERATION ERROR
1411                                    ;*  :  :  :  :  :  :  ENDIF
1412                                    ;*  :  :  :  :  :  ELSE-IS BYTE IN WRTDAT AN 'ACRC' CHARACTER?
1413                                    ;*  :  :  :  :  :  :  IF IS AN 'ACRC' BYTE
1414                                    ;*  :  :  :  :  :  :  :  THEN-CHECK ACTUAL 'ACRC' WITH SIMULATED 'ACRC'
1415                                    ;*  :  :  :  :  :  :  :  :  IF OK
1416                                    ;*  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1417                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT 'ACRC' GENERATION ERROR
1418                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1419                                    ;*  :  :  :  :  :  :  :  :  IF PARITY BIT ON 'ACRC' IS OK
1420                                    ;*  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1421                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON 'ACRC' BYTE
1422                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1423                                    ;*  :  :  :  :  :  :  :  ELSE-CONTINUE
1424                                    ;*  :  :  :  :  :  :  ENDIF
1425                                    ;*  :  :  :  :  :  :  IF BYTE ASSEMBLED IS A 'CRC' BYTE
1426                                    ;*  :  :  :  :  :  :  :  THEN-CHECK ACTUAL 'CRC' WITH SIMULATED 'CRC'
1427                                    ;*  :  :  :  :  :  :  :  :  IF SAME
```

```
1428                                    ;*  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1429                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT 'CRC' GENERATION ERROR
1430                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1431                                    ;*  :  :  :  :  :  :  :  :  IF PARITY BIT ON 'CRC' IS OK
1432                                    ;*  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1433                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON 'CRC' BYTE
1434                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1435                                    ;*  :  :  :  :  :  :  :  :  IF RESIDUAL BYTE DETECTED BEFORE 'CRC'
1436                                    ;*  :  :  :  :  :  :  :  :  :  THEN-REPORT ERROR
1437                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1438                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1439                                    ;*  :  :  :  :  :  :  :  :  IF 'ACRC' NOT DETECTED YET
1440                                    ;*  :  :  :  :  :  :  :  :  :  THEN-REPORT ERROR
1441                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1442                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1443                                    ;*  :  :  :  :  :  :  :  ELSE-CONTINUE
1444                                    ;*  :  :  :  :  :  :  :  ENDIF
1445                                    ;*  :  :  :  :  :  :  :  IF BYTE ASSEMBLED IS A RESIDUAL CHARACTER
1446                                    ;*  :  :  :  :  :  :  :  :  THEN-CHECK TO SEE IF 'ACRC' SENT
1447                                    ;*  :  :  :  :  :  :  :  :  :  IF NO 'ACRC' BYTE
1448                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-REPORT ERROR
1449                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1450                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1451                                    ;*  :  :  :  :  :  :  :  :  :  IF ALL EXPECTED 'CRC' BYTES DETECTED
1452                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1453                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1454                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1455                                    ;*  :  :  :  :  :  :  :  :  :  IF RESIDUAL BYTE SAME AS EXPECTED
1456                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1457                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1458                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1459                                    ;*  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1460                                    ;*  :  :  :  :  :  :  :  ENDIF
1461                                    ;*  :  :  :  :  :  :  :  BYTE ASSEMBLED MUST BE A DATA BYTE FROM WMC
1462                                    ;*  :  :  :  :  :  :  :  UPDATE BYTE COUNTER
1463                                    ;*  :  :  :  :  :  :  :  IF THIS BYTE EXPECTED TO BE A 'PAD'
1464                                    ;*  :  :  :  :  :  :  :  :  THEN-UPDATE # RECEIVED 'PADS'
1465                                    ;*  :  :  :  :  :  :  :  :  :  IF PAD BYTE=0
1466                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1467                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1468                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1469                                    ;*  :  :  :  :  :  :  :  :  :  IF PARITY ON 'PAD' IS OK
1470                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1471                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1472                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1473                                    ;*  :  :  :  :  :  :  :  :  :  IF ALL EXPECTED PADS CHECKED
1474                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CHECK PAD COUNTER
1475                                    ;*  :  :  :  :  :  :  :  :  :  :  :  IF PADCNT=0
1476                                    ;*  :  :  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1477                                    ;*  :  :  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1478                                    ;*  :  :  :  :  :  :  :  :  :  :  :  ENDIF
1479                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1480                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1481                                    ;*  :  :  :  :  :  :  :  :  :  IF BYTE EXPECTED TO BE A DATA BYTE
```

```
                                ;*  :  :  :  :  :  :  :  :  :  THEN-CHECK ACTUAL WITH EXPECTED DATA
                                ;*  :  :  :  :  :  :  :  :  :  :  IF SAME
                                ;*  :  :  :  :  :  :  :  :  :  .  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT WMC ASSEMBLY ERROR
                                ;*  :  :  :  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  :  :  :  :  IF PARITY BIT ON DATA ASSEMBLED IS OK
                                ;*  :  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY DETECTED
                                ;*  :  :  :  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  :  REPORT 'XFER' SET WITH NO APPARANT REASON
                                ;*  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  ENDIF
                                ;*  :  :  :  ENDIF
                                ;*  :  :  ENDIF
                                ;*  :  :  DO UNTIL ALL BYTES ASSEMBLED AND CHECKED FOR CURRENT PACKING MODE
                                ;*  :  ENDDO
                                ;*  :  CHECK BYTE COUNTER TO SEE IF=0
                                ;*  :  IF BYTE COUNTER=0
                                ;*  :  :  THEN-UPDATE FORMAT NUMBER FOR NEXT PASS
                                ;*  :  :  ELSE-REPORT FAILURE OF BYTE COUNTER TO GO ZERO WHEN EXPECTED
                                ;*  :  ENDIF
                                ;*  :  DO UNTIL FORMAT NUMBER=7 (ILLEGAL #)
                                ;*  ENDDO
                                ;*ENDTST
1509  4300                      SE
  (1)                           ;*****************************************************************************
  (1)                           ;*ERRORS
  (1)                           ;*------
1510                            ;*WMC4 MICRO TEST 01
1511                            ;*WMC4 MICRO ERROR 01
1512                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1513                            ;*M8959
1514                            ;*WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1515                            ;*TEST ABORTED!
1516                            ;*
1517                            ;*WMC4 MICRO TEST 01
1518                            ;*WMC4 MICRO ERROR 02
1519                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1520                            ;*M8959
1521                            ;*WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1522                            ;*TEST ABORTED'
1523                            ;*
1524                            ;*WMC4 MICRO TEST 01
1525                            ;*WMC4 MICRO ERROR 03
1526                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1527                            ;*M8959
1528                            ;*WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1529                            ;*TEST ABORTED.
1530                            ;*
1531                            ;*WMC4 MICRO TEST 01
1532                            ;*WMC4 MICRO ERROR 04
```

```
1533                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1534                            ;*M8959
1535                            ;*WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1536                            ;*
1537                            ;*WMC4 MICRO TEST 01
1538                            ;*WMC4 MICRO ERROR 05
1539                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1540                            ;*M8959
1541                            ;*WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1542                            ;*BYTE-SCLK COUNT = LLL
1543                            ;*DATA FORMAT = MM
1544                            ;*SKIP COUNT  = NN
1545                            ;*TEST ABORTED!
1546                            ;*
1547                            ;*WMC4 MICRO TEST 01
1548                            ;*WMC4 MICRO ERROR 06
1549                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1550                            ;*M8959
1551                            ;*WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1552                            ;*BYTE-SCLK COUNT = LLL
1553                            ;*DATA FORMAT = MM
1554                            ;*SKIP COUNT  - NN
1555                            ;*TEST ABORTED!
1556                            ;*
1557                            ;*WMC4 MICRO TEST 01
1558                            ;*WMC4 MICRO ERROR 07
1559                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1560                            ;*M8959
1561                            ;*PAD CHARACTER BYTE INCORRECT
1562                            ;*ACTUAL = NNNN
1563                            ;*EXPECTED = NNNN
1564                            ;*
1565                            ;*WMC4 MICRO TEST 01
1566                            ;*WMC4 MICRO ERROR 10
1567                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1568                            ;*M8959
1569                            ;*PARITY BIT ON PAD SHOULD =0 AND WAS  1
1570                            ;*ACTUAL - NNNN
1571                            ;*
1572                            ;*WMC4 MICRO TEST 01
1573                            ;*WMC4 MICRO ERROR 11
1574                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1575                            ;*M8959
1576                            ;*PARITY BIT ON PAD WAS  0 AND SHOULD BE -1
1577                            ;*ACTUAL = NNNN
1578                            ;*
1579                            ;*WMC4 MICRO TEST 01
1580                            ;*WMC4 MICRO ERROR 12
1581                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1582                            ;*M8959
1583                            ;*WMC PAD COUNTER NOT = -1 AFTER LAST PAD CHAR RECEIVED
1584                            ;*ACTUAL = NNNN
1585                            ;*BYTE/SCLK COUNT NUMBER - LLL
1586                            ;*
```

```
1587                            ;*WMC4 MICRO TEST 01
1588                            ;*WMC4 MICRO ERROR 13
1589                            ;*WMC4-WMC GCR CHARACTER GENFRATION TEST
1590                            ;*M8959
1591                            ;*WMC SET 'XFER' ONLY-AFTER ALL DATA AND PAD CHARACTERS OUTPUT
1592                            ;*
1593                            ;*WMC4 MICRO TEST 01
1594                            ;*WMC4 MICRO ERROR 14
1595                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1596                            ;*M8959
1597                            ;*WMC BYTE ASSEMBLY FAULT-BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
1598                            ;*ACTUAL = NNNN
1599                            ;*EXPECTED = NNNN
1600                            ;*BYTE-SCLK COUNT = LLL
1601                            ;*DATA FORMAT = MM
1602                            ;*SKIP COUNT  = NN
1603                            ;*
1604                            ;*WMC4 MICRO TEST 01
1605                            ;*WMC4 MICRO ERROR 15
1606                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1607                            ;*M8959
1608                            ;*PARITY BIT ON DATA SHOULD =0 AND WAS =1
1609                            ;*ACTUAL = NNNN
1610                            ;*
1611                            ;*WMC4 MICRO TEST 01
1612                            ;*WMC4 MICRO ERROR 16
1613                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1614                            ;*M8959
1615                            ;*PARITY BIT ON DATA WAS -0 AND SHOULD BE -1
1616                            ;*ACTUAL = NNNN
1617                            ;*
1618                            ;*WMC4 MICRO TEST 01
1619                            ;*WMC4 MICRO ERROR 17
1620                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1621                            ;*M8959
1622                            ;*WMC BYTE COUNTER EXPECTED TO BE = 0-ALL DATA BYTES RECEIVED FROM WMC
1623                            ;*ACTUAL = NNNN
1624                            ;*BYTE-SCLK COUNT = LLL
1625                            ;*DATA FORMAT = MM
1626                            ;*SKIP COUNT  = NN
1627                            ;*
1628                            ;*WMC4 MICRO TEST 01
1629                            ;*WMC4 MICRO ERROR 20
1630                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1631                            ;*M8959
1632                            ;*SPECIAL TERMINATING 'CRC' CHARACTER INCORRECT
1633                            ;*ACTUAL = NNNN
1634                            ;*EXPECTED = NNNN
1635                            ;*
1636                            ;*WMC4 MICRO TEST 01
1637                            ;*WMC4 MICRO ERROR 21
1638                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1639                            ;*M8959
1640                            ;*WMC BYTE COUNT AT END OF XFR WAS WRONG-SHOULD BE = 361 OCTAL
```

```
1641                                    ;*ACTUAL = NNNN
1642                                    ;*
1643                                    ;*WMC4 MICRO TEST 01
1644                                    ;*WMC4 MICRO ERROR 22
1645                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1646                                    ;*M8959
1647                                    ;*WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
1648                                    ;*ACTUAL = NNNN
1649                                    ;*
1650                                    ;*WMC4 MICRO TEST 01
1651                                    ;*WMC4 MICRO ERROR 23
1652                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1653                                    ;*M8959
1654                                    ;*WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED-ACTUAL = 'XFER' COUNT
1655                                    ;*ACTUAL = NNNN
1656                                    ;*BYTE-SCLK COUNT = LLL
1657                                    ;*DATA FORMAT = MM
1658                                    ;*SKIP COUNT  = NN
1659                                    ;*
1660                                    ;*WMC4 MICRO TEST 01
1661                                    ;*WMC4 MICRO ERROR 24
1662                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1663                                    ;*M8959
1664                                    ;*WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED
1665                                    ;*ACTUAL = NNNN
1666                                    ;*EXPECTED = NNNN
1667                                    ;*BYTE-SCLK COUNT = LLL
1668                                    ;*DATA FORMAT = MM
1669                                    ;*SKIP COUNT  = NN
1670                                    ;*
1671                                    ;*WMC4 MICRO TEST 01
1672                                    ;*WMC4 MICRO ERROR 25
1673                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1674                                    ;*M8959
1675                                    ;*PARITY BIT ON ECC SHOULD -0 AND WAS =1
1676                                    ;*ACTUAL = NNNN
1677                                    ;*
1678                                    ;*WMC4 MICRO TEST 01
1679                                    ;*WMC4 MICRO ERROR 26
1680                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1681                                    ;*M8959
1682                                    ;*PARITY BIT ON ECC WAS =0 AND SHOULD BE =1
1683                                    ;*ACTUAL = NNNN
1684                                    ;*
1685                                    ;*WMC4 MICRO TEST 01
1686                                    ;*WMC4 MICRO ERROR 27
1687                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1688                                    ;*M8959
1689                                    ;*WMC PRESENTED RESIDUAL CHARACTER BEFORE ACRC CHARACTER
1690                                    ;*
1691                                    ;*WMC4 MICRO TEST 01
1692                                    ;*WMC4 MICRO ERROR 30
1693                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1694                                    ;*M8959
```

```
                        ;*WMC PRESENTED CRC CHARACTER BEFORE ACRC CHARACTER
                        ;*
                        ;*WMC4 MICRO TEST 01
                        ;*WMC4 MICRO ERROR 31
                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                        ;*M8959
                        ;*WMC PRESENTED ACRC CHARACTER MORE THAN ONCE
                        ;*
                        ;*WMC4 MICRO TEST 01
                        ;*WMC4 MICRO ERROR 32
                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                        ;*M8959
                        ;*WMC FAILED TO SEND PROPER # PAD CHAR. IN RESIDUAL GROUP
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*BYTE/SCLK COUNT NUMBER = LLL
                        ;*
                        ;*WMC4 MICRO TEST 01
                        ;*WMC4 MICRO ERROR 33
                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                        ;*M8959
                        ;*WMC FAILED TO SEND ALL DATA BYTES IN RESIDUAL GROUP
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*WMC4 MICRO TEST 01
                        ;*WMC4 MICRO ERROR 34
                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                        ;*M8959
                        ;*WMC 'ACRC' DATA BYTE NOT SAME AS CALCULATED 'ACRC' DATA BYTE
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*WMC4 MICRO TEST 01
                        ;*WMC4 MICRO ERROR 35
                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                        ;*M8959
                        ;*PARITY BIT ON ACRC SHOULD -0 AND WAS =1
                        ;*ACTUAL = NNNN
                        ;*
                        ;*WMC4 MICRO TEST 01
                        ;*WMC4 MICRO ERROR 36
                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                        ;*M8959
                        ;*PARITY BIT ON ACRC WAS =0 AND SHOULD BE =1
                        ;*ACTUAL = NNNN
                        ;*
                        ;*WMC4 MICRO TEST 01
                        ;*WMC4 MICRO ERROR 37
                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                        ;*M8959
                        ;*WMC SET RESIDUAL CHARACTER BEFORE CRC CHARACTERS WERE ALL SENT
                        ;*
                        ;*WMC4 MICRO TEST 01
```

```
1749                    ;*WMC4 MICRO ERROR 40
1750                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1751                    ;*M8959
1752                    ;*WMC FAILED TO SEND ACRC CHARACTER BEFORE CRC CHARACTER
1753                    ;*
1754                    ;*WMC4 MICRO TEST 01
1755                    ;*WMC4 MICRO ERROR 41
1756                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1757                    ;*M8959
1758                    ;*WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
1759                    ;*ACTUAL = NNNN
1760                    ;*EXPECTED = NNNN
1761                    ;*
1762                    ;*WMC4 MICRO TEST 01
1763                    ;*WMC4 MICRO ERROR 42
1764                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1765                    ;*M8959
1766                    ;*PARITY BIT ON CRC SHOULD -0 AND WAS =1
1767                    ;*ACTUAL = NNNN
1768                    ;*
1769                    ;*WMC4 MICRO TEST 01
1770                    ;*WMC4 MICRO ERROR 43
1771                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1772                    ;*M8959
1773                    ;*PARITY BIT ON CRC WAS =0 AND SHOULD BE -1
1774                    ;*ACTUAL = NNNN
1775                    ;*
1776                    ;*WMC4 MICRO TEST 01
1777                    ;*WMC4 MICRO ERROR 44
1778                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1779                    ;*M8959
1780                    ;*WMC FAILED TO SEND THE 'ACRC' CHARACTER BEFORE RESIDUAL CHARACTER WAS PRESENTED
1781                    ;*
1782                    ;*WMC4 MICRO TEST 01
1783                    ;*WMC4 MICRO ERROR 45
1784                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1785                    ;*M8959
1786                    ;*WMC FAILED TO PRESENT CORRECT # CRC CHARACTERS IN CRC GROUP-ACTUAL = # RECEIVED
1787                    ;*ACTUAL = NNNN
1788                    ;*EXPECTED = NNNN
1789                    ;*
1790                    ;*WMC4 MICRO TEST 01
1791                    ;*WMC4 MICRO ERROR 46
1792                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1793                    ;*M8959
1794                    ;*WMC ACTUAL RESIDUAL DATA BYTE PRESENTED NOT SAME AS EXPECTED RESIDUAL CHAR
1795                    ;*ACTUAL = NNNN
1796                    ;*EXPECTED = NNNN
1797                    ;*
1798                    ;*WMC4 MICRO TEST 01
1799                    ;*WMC4 MICRO ERROR 47
1800                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1801                    ;*M8959
1802                    ;*PARITY BIT ON RESIDUAL SHOULD -0 AND WAS -1
```

```
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC4 MICRO TEST 01
                                        ;*WMC4 MICRO ERROR 50
                                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                                        ;*M8959
                                        ;*PARITY BIT ON RESIDUAL WAS =0 AND SHOULD BE =1
                                        ;*ACTUAL = NNNN
4300                                    $
 (1)                                    ; ************************************************************************************

                                        TST1:   TESTX   @1
 (1) 4300 3E 01          7.0                            MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302 CD 03 28      18.0                            CALL    TSET            ;SETUP THE TEST
                                        ;%WMC4-WMC GCR CHARACTER GENERATION TEST
                                        ;&M8959

                                        ;INIT FORMAT CODE COUNTER TO 5 - FORMAT CODE = IMAGE MODE

4305 3E 05              7.0     TST1ST: MVI     A,@5            ;LOAD THE IMAGE MODE FORMAT CODE
4307 32 89 4A          13.0             STA     FORMT           ;SAVE FORMAT CODE
430A AF                 4.0             XRA     A
430B 32 88 4A          13.0             STA     EABCNT          ;CLEAR THE LOOP CONTROL
430E 32 A7 4A          13.0             STA     TPBCNT          ;CLEAR THE TEST PASS BYTE COUNT

                                        ;HERE TO UPDATE THE NUMBER OF BYTES TO USE THIS TEST PASS

4311 3A A7 4A          13.0     TST1NBC: LDA    TPBCNT          ;GET THE BYTE COUNT
4314 3C                 4.0             INR     A
4315 32 A7 4A          13.0             STA     TPBCNT          ;SAVE THE NEW BYTE COUNT TO USE
                                        ;INIT COUNTER CONTROL FOR BYTE COUNTER AND ECODE COUNTER

                                        ;RESTART WRITE MICROCONTROLLER ROM PROGRAM

4318 3E FF              7.0     TST1L:  MVI     A,@377          ;-1 TO OFFSET NUMBER
431A 32 8A 4A          13.0             STA     DPNUM           ;INIT DATA PATTERN OFFSET SO 1ST WILL BE '0'
431D 3E 01              7.0             MVI     A,W.RST         ;GET 'RST' BIT
431F D3 D3             10.0             OUT     WMCCTL          ;RESTART WMC
4321 3E 05              7.0             MVI     A,SSTEP         ;SET THE SYSTEM TO SINGLE STEP MODE
4323 32 84 4A          13.0             STA     CCTLWD          ;SAVE THE CONTROL WORD
4326 D3 F0             10.0             OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
4328 CD 27 4B          18.0             CALL    CLKSYS
432B CD 27 4B          18.0             CALL    CLKSYS
432E AF                 4.0             XRA     A               ;FINISH THE WMC RESTART
432F D3 D3             10.0             OUT     WMCCTL
4331 3E 30              7.0             MVI     A,@60
4333 D3 D7             10.0             OUT     CNTCTL          ;SELECT BYTE COUNTER
4335 3E 70              7.0             MVI     A,@160
4337 D3 D7             10.0             OUT     CNTCTL          ;SELECT THE PAD COUNTER
4339 3E B0              7.0             MVI     A,@260
433B D3 D7             10.0             OUT     CNTCTL          ;SELECT ECODE COUNTER

                                        ;CLEAR ERROR CODE COUNTER
433D                                            CLRECT          ;CLEAR ECODE COUNTER
```

```
(1)  433D  AF          4.0                    XRA   A              ;CLEAR THE ACCUMULATOR
(1)  433E  D3  D6      10.0                   OUT   ERRCNT         ;CLEAR BITS 7-0
(1)  4340  D3  D6      10.0                   OUT   ERRCNT         ;CLEAR BITS 15-8
1854
1855 4342  32  A3  4A  13.0           STA   RESCNT          ;CLEAR 'RESIDUAL' CHARACTER COUNTER
1856 4345  32  9F  4A  13.0           STA   RPADCT          ;CLEAR 'PAD' COUNTER
1857 4348  32  8C  4A  13.0           STA   ECCCNT          ;CLEAR 'ECC' CHECK COUNTER
1858 434B  32  8D  4A  13.0           STA   CLKCNT          ;CLEAR THE 'SCLK' COUNTER
1859 434E  32  86  4A  13.0           STA   ABCNT           ;CLEAR THE ASSEM. BYTE COUNTER
1860 4351  32  87  4A  13.0           STA   ABCNTR
1861 4354  32  9D  4A  13.0           STA   HAVACRC         ;CLEAR THE HAVE ACRC FLAG
1862 4357  32  9C  4A  13.0           STA   HAVCRC          ;CLEAR THE HAVE CRC FLAG
1863 435A  32  9E  4A  13.0           STA   HAVRCH          ;CLEAR THE HAVE RESIDUAL CHARACTER FLAG
1864 435D  32  88  4A  13.0           STA   EABCNT          ;
1865 4360  32  A5  4A  13.0           STA   CLKCRC          ;CLEAR THE EXTRA CLOCK-TO-CRC FLAG
1866
1867                        ;SET DDR CONTROL TO "OUT" FOR DATA INJECTION INTO BAL LOGIC
1868
1869 4363  3E  88      7.0            MVI   A,@210          ;SET "OUT" BITS
1870 4365  D3  DB      10.0           OUT   DDRCTL          ;LOAD DDR CONTROL
1871
1872 4367  CD  AA  4A  18.0           CALL  GENDAT          ;GENERATE THE 1ST SET OF DATA
1873 436A  11  EF  4A  10.0           LXI   D,CDATBL        ;LOAD THE ADDRESS OF "EXPECTED" DATA TABLE
1874 436D  CD  33  4B  18.0           CALL  CLECC           ;INIT THE ECC WORD - "ECCCHR"
1875 4370  CD  02  4A  18.0           CALL  CLCRC           ;INIT CRC WORD
1876 4373  CD  16  4A  18.0           CALL  CLACRC          ;INIT ACRC WORD
1877 4376  06  B4      7.0            MVI   B,180           ;CLOCK THE SYSTEM THRU THE RESTART AREA
1878 4378  CD  27  4B  18.0  T1RST:   CALL  CLKSYS         ;CLOCK THE SYSTEM
1879 437B  05          4.0            DCR   B
1880 437C  C2  78  43  10.0           JNZ   T1RST           ;STAY HERE TILL DONE COUNTING
1881
1882 437F  D5          12.0           PUSH  D               ;SAVE D & E POINTER
1883 4380  3A  A7  4A  13.0           LDA   TPBCNT          ;GET THE BYTE COUNT TO USE
1884 4383  32  23  4F  13.0           STA   BYTEL           ;SAVE THE BYTE COUNT LOW VALUE
1885 4386  AF          4.0            XRA   A
1886 4387  32  24  4F  13.0           STA   BYTEH           ;CLEAR THE BYTE COUNT HIGH VALUE
1887 438A  CD  46  00  18.0           CALL  LKMOD7          ;CALL MOD7 ROUTINE FROM OPR U-CODE
1888 438D  3A  A7  4A  13.0           LDA   TPBCNT          ;GET THE BC AGAIN
1889 4390  3D          4.0            DCR   A               ;CALC MOD32
1890 4391  E6  1F      7.0            ANI   @37             ;SAVE LOW 5 BITS
1891 4393  57          4.0            MOV   D,A             ;TEMP SAVE
1892 4394  78          4.0            MOV   A,B             ;GET MOD7 REMAINDER FROM LKMOD7
1893 4395  32  A4  4A  13.0           STA   IMRESCNT        ;SAVE THE # RESIDUAL DATA BYTES EXPECTED
1894 4398  0F          4.0            RRC                   ;POSITION THE MOD7 NUMBER
1895 4399  0F          4.0            RRC
1896 439A  0F          4.0            RRC
1897 439B  B2          4.0            ORA   D               ;MAKE THE RESID CHAR WORD
1898 439C  32  9B  4A  13.0           STA   RCHRIM          ;SAVE IT
1899 439F  D3  D1      10.0           OUT   RESCHR          ;LOAD THE RESID CHAR WORD IN WMC
1900 43A1  3A  A7  4A  13.0           LDA   TPBCNT          ;GET THE BC AGAIN
1901 43A4  90          4.0            SUB   B               ;CALC THE VALUE LOADED INTO THE BYTE COUNTER
1902 43A5  32  85  4A  13.0           STA   IMBYTC          ;SAVE FOR LATER TESTING
1903
1904 43A8  3E  06      7.0   T1RSTA:  MVI   A,6
```

```
1905  43AA  90              4.0          SUB   B                ;CALC THE PAD COUNTER VALUE
1906  43AB  32  A6  4A     13.0          STA   IMPADCT          ;SAVE THE PAD COUNT TO USE
1907  43AE  3C               4.0          INR   A                ;ADD 1 MORE FOR THE INTEL 8253 CHIP
1908  43AF  D3  D5          10.0          OUT   PADCNT           ;LOAD PAD COUNT 7-0
1909
1910  43B1  AF               4.0  T1RSTB: XRA   A
1911  43B2  D3  D5          10.0          OUT   PADCNT           ;LOAD PAD CCOUNTER 15-8
1912  43B4  3A  A6  4A      13.0          LDA   IMPADCT          ;GET PAD COUNT AGAIN
1913  43B7  21  A7  4A      10.0          LXI   H,TPBCNT         ;POINT TO BC TO USE
1914  43BA  86               7.0          ADD   M                ;CALC # TO SEE IF PADCRC BIT NEEDED
1915  43BB  3C               4.0          INR   A                ;IN THE DATACTL WORD
1916  43BC  0F               4.0          RRC                    ;LOAD THE PADCRC BIT - IF =1 THEN PAD
1917  43BD  E6  80           7.0          ANI   BIT7             ;SAVE ONLY THIS BIT
1918  43BF  32  A8  4A      13.0          STA   PDCRC            ;SAVE AS A FLAG
1919
1920  43C2  3A  A6  4A      13.0          LDA   IMPADCT          ;GET THE PAD COUNT EXPECTED
1921  43C5  3C               4.0          INR   A                ;CORRECT FOR THE EXTRA PAD BYTE
1922  43C6  32  A6  4A      13.0          STA   IMPADCT          ;SAVE THE NEW COUNT
1923
1924                                 ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1925
1926  43C9  3A  89  4A      13.0  T1RSTC: LDA   FORMT            ;GET THE FORMAT CODE
1927  43CC  CD  1E  4B      18.0          CALL  SHL4             ;POSITION IT TO BITS 6-4
1928  43CF                        ROUT    R02H                   ;ALSO SAVE IN CAS REG 02
 (1)  43CF  D3  85          10.0                OUT     R02H          ;WRITE AC INTO R02H
 (1)  43D1  7F               4.0                MOV     A,A           ;RETRY LINK
1929  43D2  47               4.0          MOV   B,A              ;SAVE THE FMT DATA
1930  43D3  3A  A8  4A      13.0          LDA   PDCRC            ;GET THE PAD CRC BIT
1931  43D6  B0               4.0          ORA   B                ;ADD IN THE FORMAT NUMBER
1932  43D7  D3  D0          10.0          OUT   DATACTL          ;LOAD DATA CONTROL WORD
1933  43D9  3E  04           7.0          MVI   A,4              ;SET RECORD COUNT TO 1
1934  43DB                        ROUT    R02L
 (1)  43DB  D3  84          10.0                OUT     R02L          ;WRITE AC INTO R02L
 (1)  43DD  7F               4.0                MOV     A,A           ;RETRY LINK
1935  43DE  D1              10.0          POP   D                ;RESET THE POINTER IN D & E
1936
1937                                 ;ENABLE WRITE MICROCONTROLLER - START OF DATA XFR - WRITE
1938
1939  43DF  3E  98           7.0  TST1A:  MVI   A,W.ENAB!W.WRITE!W.GCR  ;SELECT 'WMC EN' + 'WRITE'
1940  43E1  D3  D3          10.0          OUT   WMCCTL           ;LOAD WRITE MICROCONTROLLER CONTROL
1941  43E3  AF               4.0          XRA   A
1942  43E4  32  8B  4A      13.0          STA   ^LKON            ;CLEAR 'SCLK' ON FLAG
1943
1944  43E7  06  FF           7.0          MVI   B,@377           ;SET SINGLE STEP CLOCK 'TIMEOUT' LIMIT TO MAX POS. COUNT
1945  43E9  48               4.0          MOV   C,B              ;COPY FOR THE 'XFER' TIMEOUT COUNTER
1946
1947  43EA  CD  73  44      18.0  TST1B:  CALL  CKSCLK          ;ISSUE SINGLE STEP CLOCK
1948  43ED  DA  18  43      10.0          JC    TST1L            ;JUMP BACK IF ERROR DETECTED
1949  43F0  DB  D0          10.0          IN    WMCSTA           ;GET WMC STATUS WORD
1950  43F2  E6  20           7.0          ANI   W.XFER           ;SAVE ONLY 'XFER' BIT
1951  43F4  C2  89  44      10.0          JNZ   XFRBIT           ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
1952  43F7  3A  95  4A      13.0          LDA   SXFER            ;GET THE SOFTWARE XFER FLAG
1953  43FA  A7               4.0          ANA   A
1954  43FB  CA  07  44      10.0          JZ    TST1C            ;JUMP IF XFER NOT SET
```

```
1955  43FE  AF                    4.0          XRA     A               ;CLEAR SOFTWARE FLAG
1956  43FF  32  95  4A            13.0         STA     SXFER           ;FOR NEXT PASS
1957  4402  06  7F                7.0          MVI     B,@177          ;SETUP A WATCHDOG COUNT
1958  4404  C3  AE  44            10.0         JMP     XFRB2           ;PROCESS THE 'XFER' BIT DETECTED
1959
1960                                        ;HERE TO SEE IF 'SCLK' TIMEOUT HAPPENED
1961
1962  4407  05                    4.0    TST1C:  DCR     B               ;COUNT CLOCK CYCLE
1963  4408  C2  19  44            10.0         JNZ     TST1D           ;TIMEOUT? - JUMP IF OK
1964
1965  4408                               ERR     TST1L,TST1BC
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4408  CD  09  28            18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  440E  01                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  440F  00                         .BYTE
 (1)  4410  CD  15  28            18.0           TST1BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4413  DA  18  43            10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1966                                    ;>WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1967                                    ;<TEST ABORTED!
1968  4416  C3  18  28            10.0         JMP     TSTEND          ;TERMINATE TESTING NOW! - FATAL ERROR
1969
1970                                        ;HERE TO SEE IF 'XFER' BIT TIMEOUT HAPPENED
1971
1972  4419  0D                    4.0    TST1D:  DCR     C               ;DECREMENT 'XFER' WATCHDOG COUNTER
1973  441A  C2  EA  43            10.0         JNZ     TST1B           ;BACK TO LOOP IF OK
1974
1975  441D                               ERR     TST1L,TST1DC
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  441D  CD  09  28            18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4420  02                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4421  00                         .BYTE
 (1)  4422  CD  15  28            18.0           TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4425  DA  18  43            10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1976                                    ;>WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1977                                    ;<TEST ABORTED!
1978  4428  C3  18  28            10.0         JMP     TSTEND
1979  4428                               S
 (1)                                   ; ***************************************************************************
1980                                    ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
1981                                    ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
1982  442B                              S
 (1)                                   ; ***************************************************************************
1983
1984  442B  3A  8B  4A            13.0 SCLKBIT: LDA     CLKON           ;SEE IF 'SCLK' WAS ON
1985  442E  A7                     4.0          ANA     A               ;SET CONDITION BITS
1986  442F  C0                    12.0          RNZ                     ;EXIT IF CLOCK WAS SET-HAVE DATA ALREADY
1987  4430  CD  CA  4A            18.0          CALL    SETDDR          ;THEN LOAD THE DATA INTO THE DDR REGS
1988  4433  C5                    12.0          PUSH    B               ;SAVE REG B & C
1989  4434  06  14                7.0          MVI     B,20            ;GET A WATCHDOG COUNT
1990  4436  CD  27  4B            18.0    SCKWT:  CALL    CLKSYS          ;CLOCK THE SYSTEM - WAIT FOR 'SCLK' TO GO AWAY
1991  4439  DB  D0                10.0          IN      WMCSTA          ;GET THE WMC STATUS WORD
1992  443B  E6  20                 7.0          ANI     W.XFER          ;DID 'XFER' BIT SET?
```

```
1993  443D  CA  43  44      10.0          JZ      SCKWT1          ;NO - JUMP OVER
1994  4440  32  95  4A      13.0          STA     SXFER           ;YUP - SET THE FLAG
1995  4443  DB  C0          10.0  SCKWT1: IN      DBUSSTA         ;GET THE DATA BUS STATUS INFO
1996  4445  E6  02           7.0          ANI     T.SCLK          ;SEE IF SCLK IS GONE YET
1997  4447  CA  5E  44      10.0          JZ      SCKGON          ;YUP - PROCEED
1998  444A  05               4.0          DCR     B               ;NOPE! - DOWNCOUNT WATCHDOG
1999  444B  C2  36  44      10.0          JNZ     SCKWT           ;STAY IN LOOP TILL NO 'SCLK' OR DIE
2000  444E                               ERR     SCKWTE,SCKWTC
  (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  444E  CD  09  28      18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)        0003                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4451  03                      .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4452  00                      .BYTE
  (1)  4453  CD  15  28      18.0    SCKWT0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4456  DA  5C  44      10.0                JC      SCKWTE          ;LOOP ADDRESS IF LOOP SPECIFIED
2001                                 ;>WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
2002                                 ;<TEST ABORTED.
2003  4459  C3  18  28      10.0          JMP     TSTEND          ;EXIT IF NOT LOOPING ON ERROR
2004
2005  445C  C1                10.0  SCKWTE: POP     B
2006  445D  C9                10.0          RET
2007  445E  C1                10.0  SCKGON: POP     B               ;ALL OK - NOW RESET REG B & C
2008  445F  C2  65  44      10.0          JNZ     SCKB1           ;JMP IF NOT TIME TO RELOAD DDR
2009
2010  4462  CD  AA  4A      18.0          CALL    GENDAT          ;GENERATE NEW DDR DATA FOR INJECTION
2011
2012  4465  06  7F           7.0  SCKB1:  MVI     B,@177          ;GET NEW TIMEOUT COUNT
2013  4467  78               4.0          MOV     A,B             ;COPY TO REG A
2014  4468  32  8B  4A      13.0          STA     CLKON           ;SET SCLK 'ON' FLAG
2015  446B  3A  8D  4A      13.0          LDA     CLKCNT          ;GET THE CURRENT SCLK COUNT
2016  446E  3C               4.0          INR     A               ;UPDATE IT
2017  446F  32  8D  4A      13.0          STA     CLKCNT          ;SAVE THE NEW COUNT
2018  4472  C9              10.0          RET
```

```
2020  4473                                  S
 (1)                                        ;*****************************************************************************
2021                                        ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
2022                                        ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
2023                                        ;       THE DDR.
2024  4473                                  S
 (1)                                        ;*****************************************************************************
2025
2026  4473  CD  27  4B        18.0  CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
2027  4476  DB  C0              10.0        IN      DBUSSTA         ;GET STATUS OF DATA BUS
2028  4478  E6  02               7.0        ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
2029  447A  C2  82  44          10.0        JNZ     CKSC1           ;JUMP IF SCLK IS ON
2030  447D  AF                   4.0        XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
2031  447E  32  88  4A          13.0        STA     CLKON
2032  4481  C9                  10.0        RET                     ;EXIT
2033
2034  4482  CD  28  44          18.0  CKSC1:  CALL    SCLKBIT         ;LOAD NEW DATA INTO DDR REGS
2035  4485  DA  18  43          10.0        JC      TST1L
2036  4488  C9                  10.0        RET                     ;EXIT
2037
2038  4489                                  S
 (1)                                        ;*****************************************************************************
2039                                        ;XFRBIT -- HERE WHEN WMC STATUS INDICATES 'XFER' BIT IS SET. ISSUE 20
2040                                        ;       MORE CLOCKS THEN CHECK WRITE DATA WORD.
2041  4489                                  S
 (1)                                        ;*****************************************************************************
2042
2043  4489  DB  D0              10.0  XFRBIT: IN      WMCSTA          ;GET THE CURRENT WMC STATUS
2044  448B  32  94  4A          13.0        STA     SWMCSTA         ;SAVE IT FOR THE ECC CHECK (LATER)
2045  448E  06  14               7.0        MVI     B,20            ;ALLOW UP TO 20. WMC CLOCKS TO DROP XFER
2046  4490  CD  73  44          18.0  XFRB1:  CALL    CKSCLK          ;NEED TWO MORE CLOCK CYCLES BEFORE
2047  4493  DA  18  43          10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2048  4496  DB  D0              10.0        IN      WMCSTA          ;GET WMC STATUS
2049  4498  E6  20               7.0        ANI     W.XFER          ;SAVE THE XFER BIT
2050  449A  CA  A6  44          10.0        JZ      XFRGON          ;OK TO CONTINUE IF XFER IS GONE
2051  449D  05                   4.0        DCR     B               ;DECREMENT WATCHDOG COUNTER
2052  449E  C2  90  44          10.0        JNZ     XFRB1           ;LOOP TIL -0 OR XFER IS GONE
2053  44A1                                  ERR     TST1L,XFRGON
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44A1  CD  09  28          18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                             MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A4  04                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A5  00                               .BYTE
 (1)  44A6  CD  15  28          18.0  XFRGON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A9  DA  18  43          10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2054                                        ;>WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
2055                                        ;HERE TO LOOK FOR 'XFER' TO SET FOR THE 2ND NIBBLE OF THIS BYTE
2056
2057  44AC  06  7F               7.0        MVI     B,@177
2058  44AE  CD  73  44          18.0  XFRB2: CALL    CKSCLK          ;SEE IF SCLK IS SET (ALSO CLOCK)
2059  44B1  DA  18  43          10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2060  44B4  DB  D0              10.0        IN      WMCSTA          ;GET WMC STATUS
2061  44B6  E6  20               7.0        ANI     W.XFER          ;SAVE ONLY XFER BIT
2062  44B8  C2  CD  44          10.0        JNZ     XFRB3           ;JUMP IF XFER IS SET - OK
```

```
2063  44BB  05                4.0          DCR     B                ;LOOK FOR TIMEOUT
2064  44BC  C2  AE  44        10.0          JNZ     XFRB2            ;GO BACK TO LOOP IF OK
2065  44BF                                  ERR     TST1L,XFRB2C,3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44BF  CD  09  28        18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0005                                    MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C2  05                                      .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44C3  03                                      .BYTE   3
 (1)  44C4  CD  15  28        18.0          XFRB2C::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44C7  DA  18  43        10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
2066                                    ;>WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
2067                                    ;<TEST ABORTED!
2068  44CA  C3  18  28        10.0          JMP     TSTEND
2069
2070                                    ;HERE TO LOOK FOR 'XFER' BIT TO GO AWAY TO INDICATE FULL BYTE IS AVAIL.
2071                                    ;FROM THE 'WRTDAT' WORD
2072
2073  44CD  06  14              7.0   XFRB3:  MVI     B,20
2074  44CF  DB  E0             10.0           IN      INTSTA           ;GET THE DATA PARITY BIT
2075  44D1  32  96  4A         13.0           STA     SINTSTA          ;SAVE FOR PARITY CHECK
2076  44D4  CD  73  44         18.0   XFRB3A: CALL    CKSCLK           ;CLOCK THE SYSTEM TILL XFER BIT GOES AWAY
2077  44D7  DA  18  43         10.0           JC      TST1L            ;JUMP IF ERROR DETECTED
2078  44DA  DB  D0             10.0           IN      WMCSTA           ;GET THE WMC STATUS
2079  44DC  E6  20              7.0           ANI     W.XFER           ;SAVE ONLY THE XFER BIT
2080  44DE  CA  F3  44         10.0           JZ      XFRB4            ;JUMP IF GONE - OK
2081  44E1  05                  4.0           DCR     B                ;DOWNCOUNT WATCHDOG
2082  44E2  C2  D4  44         10.0           JNZ     XFRB3A           ;BACK TO LOOP IF OK
2083  44E5                                   ERR     TST1L,XFRB3C,3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E5  CD  09  28        18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0006                                    MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E8  06                                      .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44E9  03                                      .BYTE   3
 (1)  44EA  CD  15  28        18.0          XFRB3C::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44ED  DA  18  43        10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
2084                                    ;>WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
2085                                    ;<TEST ABORTED!
2086  44F0  C3  18  28        10.0          JMP     TSTEND
2087                                    ;HERE TO TEST THE ASSEMBLED BYTE DATA AGAINST THE EXPECTED BYTE DATA
2088
2089  44F3  CD  27  4B         18.0   XFRB4:  CALL    CLKSYS           ;ALLOW THE BYTE COUNTER TO TICK
2090  44F6  CD  27  4B         18.0           CALL    CLKSYS           ;TO LATCH IT.
```

```
2092  44 3                              S
                                        ;*****************************************************************************
2093                                    ;XFRRDY -- ROUTINE TO DETERMINE WHAT CHARACTER IS ON WMC OUTPUT WHEN
2094                                    ;       'XFER' IS SET.  ALSO, CHECKS FOR ERRORS BASED ON BYTE COUNTER
2095                                    ;       AND PAD COUNTER.
2096  44F9                              S
                                        ;*****************************************************************************
 (1)
2097
2098  44F9  3A  94  4A        13.0  XFRRDY: LDA     SWMCSTA         ;GET WMC STATUS WORD
2099  44FC  E6  1E             7.0          ANI     @36             ;SAVE JUST THE SPECIAL CHAR INFO.
2100  44FE  CA  28  45        10.0          JZ      XFRB4D          ;NOTHING SPECIAL - CHECK DATA BYTE
2101
2102  4501  3A  94  4A        13.0          LDA     SWMCSTA         ;GET THE STATUS AGAIN
2103  4504  E6  10             7.0          ANI     W.ECC           ;SEE IF ITS 'ECC' CHARACTER
2104  4506  C2  32  47        10.0          JNZ     CKECC           ;JUMP IF ITS 'ECC'
2105
2106                                  ;TEST FOR 'ACRC', 'CRC', OR 'RESID' CHARACTER
2107
2108  4509  3A  8C  4A        13.0          LDA     ECCCNT          ;GET 'XFER' COUNT
2109  450C  3C                 4.0          INR     A               ;ADD 1
2110  450D  32  8C  4A        13.0          STA     ECCCNT          ;SAVE THE UPDATED COUNT
2111  4510  3A  94  4A        13.0          LDA     SWMCSTA         ;GET WMC STATUS WORD AGAIN
2112  4513  E6  04             7.0          ANI     W.ACRC          ;SEE IF 'ACRC' CHARACTER READY
2113  4515  C2  B1  47        10.0          JNZ     CKACRC          ;GO TEST 'ACRC' CHARACTER
2114
2115  4518  3A  94  4A        13.0          LDA     SWMCSTA         ;GET STATUS AGAIN
2116  451B  E6  08             7.0          ANI     W.CRC           ;SEE IF ITS 'CRC' READY BIT
2117  451D  C2  8F  48        10.0          JNZ     CKCRC           ;GO TEST 'CRC' DATA BYTE
2118
2119  4520  3A  94  4A        13.0          LDA     SWMCSTA         ;GET STATUS AGAIN
2120  4523  E6  02             7.0          ANI     W.RESID         ;SEE IF 'RESID' READY BIT (IF NOT ITS FATAL.)
2121  4525  C2  3C  49        10.0          JNZ     CKRESID         ;GO TEST 'RESID' DATA BYTE
2122
2123                                  ;HERE TO CHECK THE DATA BYTE JUST ASSEMBLED
2124
2125  4528  3A  8C  4A        13.0  XFRB4D: LDA     ECCCNT          ;GET 'XFER' COUNT
2126  452B  3C                 4.0          INR     A               ;ADD 1
2127  452C  32  8C  4A        13.0          STA     ECCCNT          ;SAVE THE UPDATED COUNT
2128  452F  E5                 12.0         PUSH    H               ;SAVE H&L
2129  4530  21  85  4A        10.0          LXI     H,IMBYTC        ;POINT TO THE AMT OF BYTES EXPECTED
2130  4533  3A  86  4A        13.0          LDA     ABCNT           ;THEN GET # BYTES ASSEMBLED BY THE WMC
2131  4536  BE                 7.0          CMP     M               ;SEE IF DONE
2132  4537  E1                 10.0         POP     H               ;RESET REG H & L
2133  4538  DA  09  46        10.0          JC      XFRB4C          ;CONTINUE TO COMPARE DATA BYTES
2134
2135
2136  453B  E5                 12.0  XFRB4R: PUSH    H               ;SAVE H&L
2137  453C  21  A4  4A        10.0          LXI     H,IMRESCNT      ;GET THE RESIDUAL DATA BYTE COUNT
2138  453F  3A  A3  4A        13.0          LDA     RESCNT          ;GET THE NUMBER OF RESIDUAL CHARACTERS TO DATE
2139  4542  BE                 7.0          CMP     M               ;SEE IF MORE RESIDUAL DATA BYTES EXPECTED
2140  4543  E1                 10.0         POP     H               ;RESTORE H&L
2141  4544  CA  51  45        10.0          JZ      CKPAD           ;JUMP IF NO MORE DATA BYTES EXPECTED
2142  4547  3A  A3  4A        13.0          LDA     RESCNT          ;GET THE RESIDUAL CHARACTERS COUNT
2143  454A  3C                 4.0          INR     A               ;UPDATE FOR THIS CHARACTER
```

```
2144  454B  32  A3  4A      13.0          STA     RESCNT          ;SAVE IT
2145  454E  C3  09  46      10.0          JMP     XFRB4C          ;GO CHECK THE RESIDUAL DATA BYTE
2146
2147                                ;HERE TO CHECK THE PAD CHARACTERS EXPECTED
2148
2149  4551  3A  A6  4A      13.0  CKPAD:  LDA     IMPADCT         ;GET THE PAD COUNT EXPECTED
2150  4554  47                     4.0          MOV     B,A             ;TEMP STORE
2151  4555  3A  A8  4A      13.0          LDA     PDCRC           ;GET THE PAD CRC FLAG
2152  4558  A7                     4.0          ANA     A
2153  4559  C2  5D  45      10.0          JNZ     CKPADA          ;JUMP OVER IF EXTRA PAD EXPECTED
2154  455C  05                     4.0          DCR     B               ;CORRECT THE PAD COUNT FOR PHANTOM XFR
2155
2156  455D  3A  9F  4A      13.0  CKPADA: LDA     RPADCT          ;GET THE RECEIVED PAD COUNT #
2157  4560  B8                     4.0          CMP     B               ;SAME AS EXPECTED (TIME FOR PHANTOM XFR)
2158  4561  CA  CC  45      10.0          JZ      CKHACK          ;GO CHECK FOR THE PHANTOM 'XFR'
2159  4564  3A  9F  4A      13.0          LDA     RPADCT          ;GET THE PAD CHARACTERS COUNT
2160  4567  3C                     4.0          INR     A               ;UPDATE FOR THIS CHARACTER
2161  4568  32  9F  4A      13.0          STA     RPADCT          ;SAVE IT
2162  456B  AF                     4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2163  456C                                ROUT    EDATA           ;STORE THE EXPECTED PAD CHARACTER
 (1)  456C  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  456E  7F                     4.0              MOV     A,A             ;RETRY LINK
2164  456F  DB  D3            10.0          IN      WRTDAT          ;GET THE ACTUAL PAD CHARACTER
2165  4571                                ROUT    ADATA           ;STORE THE ACTUAL PAD CHARACTER
 (1)  4571  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4573  7F                     4.0              MOV     A,A             ;RETRY LINK
2166  4574  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2167  4575  CA  7D  45      10.0          JZ      PADCN0          ;CONTINUE IF ZERO
2168  4578                                ERRB    TST1L,PADCN0
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4578  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  457B  07                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  457C  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  457D  CD  15  28      18.0      PADCN0::            CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4580  DA  18  43      10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2169                          ;>PAD CHARACTER BYTE INCORRECT
2170                          ;HERE TO CHECK THE PARITY BIT ON THE PAD ASSEMBLED
2171
2172  4583  DB  D3            10.0          IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2173  4585                                ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  4585  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4587  7F                     4.0              MOV     A,A             ;RETRY LINK
2174  4588  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2175  4589  E4  9B  45      18.0          CPO     PADPO           ;CHECK FOR ODD PARITY
2176  458C  DA  18  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2177  458F  DB  D3            10.0          IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2178  4591  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2179  4592  EC  AF  45      18.0          CPE     PADPE           ;CHECK FOR PARITY EVEN - BIT =1
2180  4595  DA  18  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2181  4598  C3  C1  45      10.0          JMP     CKPAD1          ;ON TO THE NEXT SECTION....
2182
2183                                ;HERE TO CHECK FOR PARITY ODD - BIT =0
2184
```

```
2185  459B  3A 96 4A    13.0  PADPO:  LDA     SINTSTA          ;GET THE PARITY BIT (IN BIT 0)
2186  459E  E6 10          7.0          ANI     WDR.P            ;SAVE THE PARITY BIT
2187  45A0  CA AE 45       10.0         JZ      PADPOE           ;EXIT IF OK
2188  45A3                               ERRA    PADPOE,PADPOC
  (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
  (1)  45A3  CD 0F 28       18.0                    CALL    ERLPA            ;PROCESS ERROR - DO 2.3
  (1)       0008                                    MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  45A6  08                                     .BYTE   MSGN             ;MESSAGE NUMBER ID
  (1)  45A7  00                                     .BYTE                    ;PRINT ROUTINE NUMBER
  (1)  45A8  CD 15 28       18.0            PADPOC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
  (1)  45AB  DA AE 45       10.0                    JC      PADPOE           ;LOOP ADDRESS IF LOOP SPECIFIED
2189                               ;>PARITY BIT ON PAD SHOULD =0 AND WAS =1
2190  45AE  C9              10.0  PADPOE: RET
2191
2192                               ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2193
2194  45AF  3A 96 4A        13.0  PADPE:  LDA     SINTSTA          ;GET THE BIT
2195  45B2  E6 10           7.0          ANI     WDR.P            ;SAVE THE PARITY BIT
2196  45B4  C0              12.0         RNZ
2197  45B5                               ERRA    PADPEE,PADPEC
  (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
  (1)  45B5  CD 0F 28       18.0                    CALL    ERLPA            ;PROCESS ERROR - DO 2.3
  (1)       0009                                    MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  45B8  09                                     .BYTE   MSGN             ;MESSAGE NUMBER ID
  (1)  45B9  00                                     .BYTE                    ;PRINT ROUTINE NUMBER
  (1)  45BA  CD 15 28       18.0            PADPEC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
  (1)  45BD  DA C0 45       10.0                    JC      PADPEE           ;LOOP ADDRESS IF LOOP SPECIFIED
2198                               ;>PARITY BIT ON PAD WAS =0 AND SHOULD BE =1
2199  45C0  C9              10.0  PADPEE: RET
2200  45C1  DB D3           10.0  CKPAD1: IN      WRTDAT           ;GET THE PAD CHARACTER AGAIN
2201  45C3  CD 3D 4B        18.0         CALL    GENECC           ;GO INCLUDE THE PAD CHARACTER IN THE ECC
2202  45C6  CD D8 49        18.0         CALL    GENCRC           ;GO INCLUDE THE PAD CHARACTER IN THE CRC
2203  45C9  C3 89 48        10.0         JMP     SCHREX           ;CONTINUE WITH THE TEST
2204
2205                               ;HERE IF A 'XFER' DATA BYTE WAS DETECTED AND ALL DATA BYTES WERE
2206                               ;RECEIVED AND ALL PAD BYTES REVEIVED...I.E. WE ARE ABOUT TO GIVE
2207                               ;AN EXTRA CLOCK TO THE CRC GENERATOR (CRC GROUP HAS 9 CHAR).
2208
2209  45CC  3A A5 4A        13.0  CKHACK: LDA     CLKCRC           ;GET THE EXTRA CLOCK-TO-CRC FLAG
2210  45CF  A7              4.0          ANA     A                ;SEE IF HERE BEFORE
2211  45D0  C2 FB 45        10.0         JNZ     XFRERR           ;YES - THEN NO MORE PHANTOM 'XFR'S' ALLOWED
2212  45D3  3E FF           7.0          MVI     A,@377           ;NO - THEN ALLOW ONLY THE ONE
2213  45D5  32 A5 4A        13.0         STA     CLKCRC           ;UPDATE THE RECEIVED FLAG
2214  45D8  3A 8C 4A        13.0         LDA     ECCCNT           ;GET THE ECC CHAR. COUNTER
2215  45DB  3D              4.0          DCR     A                ;ADJUST FOR ADDITION OF THE PHANTOM 'XFR'
2216  45DC  32 8C 4A        13.0         STA     ECCCNT           ;RESTORE ADJUSTED COUNT
2217
2218                               ;CHECK TO SEE IF PAD COUNTER = -1 AFTER LAST PAD RECEIVED
2219
2220  45DF  DB D5           10.0         IN      PADCNT           ;GET LOW BYTE OF COUNTER
2221  45E1  47              4.0          MOV     B,A              ;TEMP STORAGE
2222  45E2  DB D5           10.0         IN      PADCNT           ;GET HIGH BYTE (THROW AWAY)
2223  45E4  78              4.0          MOV     A,B              ;RESET 'A'
2224  45E5  FE FF           7.0          CPI     @377             ;SEE IF PHANTOM CAUSED COUNT OF -1
```

```
2225  45E7  CA  F2  45      10.0          JZ      XFRRC1          ;JUMP IF COUNTER = -1 (EXPECTED)
2226  45EA                                ROUT    ADATA           ;ERROR - SAVE PAD COUNTER VALUE
 (1)  45EA  D3  94          10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45EC  7F               4.0                   MOV     A,A             ;RETRY LINK
2227  45ED                                ERRA    TST1L,XFRRC1,1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  45ED  CD  0F  28      18.0                   CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000A                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45F0  0A                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45F1  01                                     .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  45F2  CD  15  28      18.0           XFRRC1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45F5  DA  18  43      10.0                   JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2228                                ;>WMC PAD COUNTER NOT = -1 AFTER LAST PAD CHAR RECEIVED
2229  45F8  C3  89  48      10.0          JMP     SCHREX          ;CONTINUE TO CLOCK THE WMC
2230  45FB                          XFRERR: ERR     TST1L,XFB4C
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45FB  CD  09  28      18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45FE  08                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45FF  00                                     .BYTE
 (1)  4600  CD  15  28      18.0           XFB4C:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4603  DA  18  43      10.0                   JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2231                                ;>WMC SET 'XFER' ONLY - AFTER ALL DATA AND PAD CHARACTERS OUTPUT
2232
2233  4606  C3  89  48      10.0          JMP     SCHREX          ;CONTINUE WITH THE TEST
2234
2235                                ;GET ASSEMBLED BAL DATA BYTE FROM WRITE DATA WORDS
2236
2237  4609  3A  86  4A      13.0  XFRB4C: LDA     ABCNT           ;GET THE BYTE # UNDER TEST
2238  460C  3C                    4.0          INR     A               ;ADD 1
2239  460D  32  86  4A      13.0          STA     ABCNT           ;SAVE THE BYTE NUMBER - UPDATED
2240  4610                                ROUT    R05L
 (1)  4610  D3  8A          10.0                   OUT     R05L            ;WRITE AC INTO R05L
 (1)  4612  7F               4.0                   MOV     A,A             ;RETRY LINK
2241  4613  AF                    4.0          XRA     A
2242  4614                                ROUT    R05H
 (1)  4614  D3  8B          10.0                   OUT     R05H            ;WRITE AC INTO R05H
 (1)  4616  7F               4.0                   MOV     A,A             ;RETRY LINK
2243  4617  EB                    4.0          XCHG                    ;PUT 'EXPECTED' POINTER IN H & L
2244  4618  7E                    7.0          MOV     A,M             ;GET CORRECT BAL BYTE
2245  4619                                ROUT    EDATA           ;SAVE 'EXPECTED' DATA
 (1)  4619  D3  95          10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  461B  7F               4.0                   MOV     A,A             ;RETRY LINK
2246  461C  DB  D3          10.0          IN      WRTDAT          ;GET ASSEMBLED DATA BYTE
2247  461E                                ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 (1)  461E  D3  94          10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4620  7F               4.0                   MOV     A,A             ;RETRY LINK
2248  4621  BE                    7.0          CMP     M               ;EXPECTED = ACTUAL?
2249  4622  EB                    4.0          XCHG                    ;RESET H & L REGS
2250  4623  13                    6.0          INX     D               ;POINT TO NEXT 'EXPECTED' ENTRY
2251  4624  CA  32  46      10.0          JZ      TST1XC          ;JUMP OVER ERROR IF DATA BYTE OK
2252  4627  3A  86  4A      13.0          LDA     ABCNT           ;GET THE COUNT OF ASSEM. BYTES AT FAILURE
2253  462A  32  88  4A      13.0          STA     EABCNT          ;SAVE FOR THE LOOP CONTROL
```

```
2255  462D                                        ERRB    DELOOP,TST1XC,3
 (1)                                              ;FLAG ERROR - WITH [illegible] 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  462D  CD  12  28        18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000C                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4630  0C                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4631  03                                          .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4632  CD  15  28        18.0              TST1XC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4635  DA  17  47        10.0                      JC      DELOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2256                                    ;>WMC BYTE ASSEMBLY FAULT - B [illegible]TE ASSEMBLED DOESN'T MATCH EXPECTED DATA
2257
2258                                    ;HERE TO CHECK THE PARITY BIT ON THE DATA ASSEMBLED
2259
2260  4638  DB  D3            10.0  XFRB4P: IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2261  463A                                  ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  463A  D3  94            10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  463C  7F                 4.0                      MOV     A,A             ;RETRY LINK
2262  463D  A7                 4.0                  ANA     A               ;SET THE CONDITION BITS
2263  463E  E4  50  46         18.0                  CPO     DATPO           ;CHECK FOR ODD PARITY
2264  4641  DA  18  43         10.0                  JC      TST1L           ;JUMP IF ERROR DETECTED
2265  4644  DB  D3            10.0                  IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2266  4646  A7                 4.0                  ANA     A               ;SET THE CONDITION BITS
2267  4647  EC  64  46         18.0                  CPE     DATPE           ;CHECK FOR PARITY EVEN - BIT -1
2268  464A  DA  18  43         10.0                  JC      TST1L           ;JUMP IF ERROR DETECTED
2269  464D  C3  76  46         10.0                  JMP     XFRB5           ;ON TO THE NEXT SECTION....
2270
2271                                    ;HERE TO CHECK FOR PARITY ODD - BIT =0
2272
2273  4650  3A  96  4A         13.0  DATPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2274  4653  E6  10                7.0                  ANI     WDR.P           ;SAVE THE PARITY BIT
2275  4655  CA  63  46         10.0                  JZ      DATPOE          ;EXIT IF OK
2276  4658                                        ERRA    DATPOE,DATPOC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4658  CD  0F  28        18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000D                                        MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  465B  0D                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  465C  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  465D  CD  15  28        18.0              DATPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4660  DA  63  46         10.0                      JC      DATPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2277                                    ;>PARITY BIT ON DATA SHOULD =0 AND WAS =1
2278  4663  C9                10.0  DATPOE: RET
2279
2280                                    ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2281
2282  4664  3A  96  4A         13.0  DATPE:  LDA     SINTSTA         ;GET THE BIT
2283  4667  E6  10                7.0                  ANI     WDR.P           ;SAVE THE PARITY BIT
2284  4669  C0                12.0                  RNZ
```

```
2286  466A                                          ERRA    DATPEE,DATPEC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  466A  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000E                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  466D  0E                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  466E  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  466F  CD  15  28        18.0                  DATPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4672  DA  75  46        10.0                          JC      DATPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2287                                        ;>PARITY BIT ON DATA WAS =0 AND SHOULD BE =1
2288  4675  C9                 10.0  DATPEE: RET
2289  4676  DB  D3              10.0  XFRB5:  IN      WRTDAT          ;GET THE DATA AGAIN
2290  4678  CD  3D  4B          18.0          CALL    GENECC          ;CALCULATE THE ECC
2291  467B  CD  D8  49          18.0          CALL    GENCRC          ;ADD DATA TO CRC CHAR
2292  467E  CD  1E  4A          18.0          CALL    GENACRC         ;AND TO THE ACRC CHAR
2293
2294  4681  E5                 12.0          PUSH    H
2295  4682  21  85  4A          10.0          LXI     H,IMBYTC        ;POINT TO THE AMT OF BYTES EXPECTED
2296  4685  3A  86  4A          13.0          LDA     ABCNT           ;THEN GET # BYTES ASSEMBLED BY THE WMC
2297  4688  BE                  7.0          CMP     M               ;SEE IF DONE
2298  4689  E1                 10.0          POP     H               ;RESET REG H & L
2299  468A  C2  89  48          10.0          JNZ     SCHREX          ;BACK TO LOOP IF MORE TO DO
2300
2301  468D  DB  D4              10.0          IN      BYTCNT          ;GET BYTE COUNTER 7-0
2302  468F  47                   4.0          MOV     B,A             ;TEMP STORE
2303  4690  DB  D4              10.0          IN      BYTCNT          ;GET BYTE CNT 15-8
2304  4692  78                   4.0          MOV     A,B
2305  4693  A7                   4.0          ANA     A
2306  4694  CA  9F  46          10.0          JZ      XFRB5C          ;JUMP IF BYTE COUNT OK
2307  4697                                    ROUT    ADATA           ;STORE THE ACTUAL BYTE COUNTER 7-0
 (1)  4697  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4699  7F                   4.0                  MOV     A,A             ;RETRY LINK
2308  469A                                    ERRA    TST1L,XFRB5C,3
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  469A  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000F                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  469D  0F                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  469E  03                                              .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  469F  CD  15  28        18.0                  XFRB5C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46A2  DA  18  43        10.0                          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2309                                        ;>WMC BYTE COUNTER EXPECTED TO BE = 0 - ALL DATA BYTES RECEIVED FROM WMC
2310
2311  46A5  C3  89  48          10.0          JMP     SCHREX          ;BACK TO CLOCK LOOP FOR OTHER CHARACTERS
```

```
2313  46A8                                    S
 (1)                                          ; ***************************************************************************
2314                                          ;HERE IF ALL DATA, ECC, CRC, ACRC, AND RESID CHAR TESTED
2315  46A8                                    S
 (1)                                          ; ***************************************************************************
2316
2317  46A8  DB  D3            10.0  QUIT:     IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
2318  46AA                                    ROUT    ADATA
 (1)  46AA  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46AC  7F                   4.0                    MOV     A,A             ;RETRY LINK
2319  46AD  21  98  4A          10.0            LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
2320  46B0  BE                   7.0            CMP     M               ;ACTUAL=SIMULATED?
2321  46B1  CA  BD  46          10.0            JZ      QUIT1           ;JUMP IF OK
2322  46B4  7E                   7.0            MOV     A,M             ;GET SIMULATED CRC CHAR.
2323  46B5                                    ROUT    EDATA
 (1)  46B5  D3  95              10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46B7  7F                   4.0                    MOV     A,A             ;RETRY LINK
2324  46B8                                    ERRB    TST1L,QUIT1
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46B8  CD  12  28          18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0010                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46BB  10                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46BC  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46BD  CD  15  28          18.0            QUIT1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46C0  DA  18  43          10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2325                                          ;>SPECIAL TERMINATING 'CRC' CHARACTER INCORRECT
2326  46C3  3A  85  4A          13.0            LDA     IMBYTC          ;GET THE IMAGE BYTE COUNTER
2327  46C6  A7                   4.0            ANA     A               ;SET THE CONDITION BITS
2328  46C7  CA  E3  46          10.0            JZ      QUIT4           ;SKIP THE BYTE COUNTER CHECK IF THE BYTECOUNTER
2329                                                                    ;WAS LOADED WITH ZEROS
2330  46CA  DB  D4              10.0            IN      BYTCNT          ;GET BYTE COUNT BITS 7-0
2331  46CC  47                   4.0            MOV     B,A             ;TEMP SAVE
2332  46CD  DB  D4              10.0            IN      BYTCNT          ;GET BITS 15-8
2333  46CF  78                   4.0            MOV     A,B             ;USE ONLY 7-0 FOR THE CHECK
2334  46D0                                    ROUT    ADATA           ;SAVE THE ACTUAL BYTE COUNT
 (1)  46D0  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46D2  7F                   4.0                    MOV     A,A             ;RETRY LINK
2335  46D3  FE  F1               7.0            CPI     @361            ;SHOULD TERMINATE WITH -17 OCTAL (361)
2336  46D5  CA  DD  46          10.0            JZ      QUIT2           ;JUMP IF BYTE COUNTER OK
2337  46D8                                    ERRA    TST1L,QUIT2
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  46D8  CD  0F  28          18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0011                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46DB  11                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46DC  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46DD  CD  15  28          18.0            QUIT2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46E0  DA  18  43          10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2338                                          ;>WMC BYTE COUNT AT END OF XFR WAS WRONG - SHOULD BE = 361 OCTAL
2339
2340  46E3  DB  D0              10.0  QUIT4:   IN      WMCSTA          ;GET THE WMC STATUS
2341  46E5                                    ROUT    ADATA           ;SAVE THE ACTUAL STATUS FOR PRINTOUT
 (1)  46E5  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46E7  7F                   4.0                    MOV     A,A             ;RETRY LINK
2342  46E8  E6  40               7.0            ANI     W.DONN          ;DONE?
```

```
2343  46EA  CA  F2  46      10.0          JZ      QUIT3             ;YES - EXIT THE TEST
2344  46ED                                ERRA    TST1L,QUIT3
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  46ED  CD  0F  28      18.0                  CALL    ERLPA             ;PROCESS ERROR - DO 2.3
 (1)        0012                                  MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46F0  12                                    .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  46F1  00                                    .BYTE                     ;PRINT ROUTINE NUMBER
 (1)  46F2  CD  15  28      18.0          QUIT3:: CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46F5  DA  18  43      10.0                  JC      TST1L             ;LOOP ADDRESS IF LOOP SPECIFIED
2345                                ;>WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
2346
2347  46F8  3A  A7  4A      13.0          LDA     TPBCNT            ;GET THE CURRENT BYTE COUNT USED
2348  46FB  FE  26           7.0          CPI     DTEND-CDATBL      ;DONE WITH ALL COMBINATIONS?
2349  46FD  C2  11  43      10.0          JNZ     TST1NBC           ;JUMP IF NOT AT END OF ROPE...
2350
2351  4700                                ENDTST  TST1ST
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4700                                REQ     7                 ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4700  CD  06  28      18.0                  CALL    REQST
 (2)  4703  00                                    .BYTE                     ;DATA PATTERN NUMBER
 (2)  4704  00  00                                .WORD                     ;SYSTEM '*' COUNT
 (2)  4706  00  00                                .WORD                     ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4708  00                                    .BYTE                     ;DATA COMPARE FLAG IF =1
 (2)  4709  07                                    .BYTE   7                 ;REQUEST CODE
 (1   470A  3A  9A  4F      13.0          LDA     ITERA             ;GET ITERATION COUNT
 (1)  470D  3D           4.0          DCR     A                 ;DOWNCOUNT
 (1)  470E  32  9A  4F      13.0          STA     ITERA             ;SAVE COUNT
 (1)  4711  F2  05  43      10.0          JP      TST1ST            ;DO TEST UNTIL TILL - 0
2352  4714  C3  18  28      10.0          JMP     TSTEND            ;ALL DONE
2353
2354                                ;HERE IS THE DATA COMPARE ERROR LOOP CONTROL ROUTINE
2355
2356                                ;WILL ENABLE THE TEST TO LOOP AT THE SPECIFIED BYTE IN ERROR
2357
2358  4717  3A  94  4F      13.0  DELOOP: LDA     LPFLG             ;SEE IF LOOP FLAG STILL SET
2359  471A  A7           4.0          ANA     A
2360  471B  C2  24  47      10.0          JNZ     DELP1             ;JUMP IF LOOPING ON ERROR
2361  471E  32  88  4A      13.0          STA     EABCNT            ;NOT LOOPING - CLEAR THE FLAG
2362  4721  C3  76  46      10.0          JMP     XFRB5             ;AND EXIT
2363
2364  4724  3A  86  4A      13.0  DELP1:  LDA     ABCNT             ;GET THE CURRENT COUNT OF ASSEMBLED BYTES
2365  4727  47           4.0          MOV     B,A               ;TEMP STORE
2366  4728  3A  88  4A      13.0          LDA     EABCNT            ;GET THE BYTE COUNT AT THE DETECTED ERROR
2367  472B  B8           4.0          CMP     B                 ;SEE IF AT THE SAME BYTE COUNT
2368  472C  CA  18  43      10.0          JZ      TST1L             ;THEN JUMP TO RESTART TEST IF SAME
2369  472F  C3  76  46      10.0          JMP     XFRB5             ;ELSE, CONTINUE WITH THE TEST
```

```
                                        S
                                        ; ***************************************************************************
                                        ;CKECC -- ROUTINE TO CHECK THE 'ECC' CHARACTER PRESENTED BY THE WMC
                                        ;       LOGIC AGAINST A CACULATED ECC BYTE. WILL REPORT AN ERROR IF
                                        ;       THE 'ECC' FLAG (IN WMCSTA WORD) COMES UP BEFORE OR AFTER THE
                                        ;       8TH ASSEMBLED BYTE FROM THE WMC OR IF AN ACTUAL ECC BYTE
                                        ;       ASSEMBLY ERROR WAS DETECTED.
                                        S
                                        ; ***************************************************************************

4732    3A  8C  4A      13.0  CKECC:  LDA     ECCCNT          ;GET 'XFER' CYCLE COUNT
4735                          ROUT    ADATA
4735    D3  94          10.0          OUT     ADATA           ;WRITE AC INTO ADATA
4737    7F               4.0          MOV     A,A             ;RETRY LINK
4738    FE  07           7.0          CPI     7               ;READY FOR ECC CHAR?
473A    CA  48  47      10.0          JZ      ECCCK           ;JUMP IF EXPECTED NOW
473D                          ERRA    TST1L,CKECCC,3
                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
473D    CD  0F  28      18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
        0013                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
4740    13                    .BYTE   MSGN            ;MESSAGE NUMBER ID
4741    03                    .BYTE   3               ;PRINT ROUTINE NUMBER
4742    CD  15  28      18.0  CKECCC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
4745    DA  18  43      10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                              ;>WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED - ACTUAL - 'XFER' COUNT

                              ;HERE TO CHECK THE ACTUAL ECC BYTE FROM THE WMC LOGIC

4748    21  97  4A      10.0  ECCCK:  LXI     H,ECCCHR        ;POINT TO 'IMAGE' OF HARDWARE ECC WORD
474B    7E               7.0          MOV     A,M
474C                          ROUT    EDATA
474C    D3  95          10.0          OUT     EDATA           ;WRITE AC INTO EDATA
474E    7F               4.0          MOV     A,A             ;RETRY LINK
474F    DB  D3          10.0          IN      WRTDAT          ;GET ACTUAL ECC WORD FROM WMC
4751                          ROUT    ADATA
4751    D3  94          10.0          OUT     ADATA           ;WRITE AC INTO ADATA
4753    7F               4.0          MOV     A,A             ;RETRY LINK
4754    BE               7.0          CMP     M               ;IS IT OK?
4755    CA  5D  47      10.0          JZ      ECCCK1          ;JUMP IF SAME-AS EXPECTED
4758                          ERRB    TST1L,ECCCK1,3
                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
4758    CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
        0014                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
475B    14                    .BYTE   MSGN            ;MESSAGE NUMBER ID
475C    03                    .BYTE   3               ;PRINT ROUTINE NUMBER
475D    CD  15  28      18.0  ECCCK1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
4760    DA  18  43      10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                              ;>WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED

4763    3A  9E  4A      13.0          LDA     HAVRCH          ;GET THE ''HAVE RESID'' FLAG
4766    A7               4.0          ANA     A
4767    CA  6D  47      10.0          JZ      ECCCK2          ;JUMP IF DON'T HAVE RESID YET
476A    32  A9  4A      13.0          STA     LASTCRC         ;THIS MUST BE LAST ECC...SET EXPECT LAST CRC FLAG
```

```
2403                                        ;HERE TO CHECK THE PARITY BIT ON THE ECC ASSEMBLED
2404
2405  476D  DB  D3          10.0  ECCCK2: IN      WRTDAT          ;GET THE ECC BYTE AGAIN
2406  476F                              ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  476F  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4771  7F               4.0                MOV     A,A             ;RETRY LINK
2407  4772  A7               4.0          ANA     A               ;SET THE CONDITION BITS
2408  4773  E4  85  47       18.0         CPO     ECCPO           ;CHECK FOR ODD PARITY
2409  4776  DA  18  43       10.0         JC      TST1L           ;JUMP IF ERROR DETECTED
2410  4779  DB  D3           10.0         IN      WRTDAT          ;GET THE ECC BYTE AGAIN
2411  477B  A7               4.0          ANA     A               ;SET THE CONDITION BITS
2412  477C  EC  99  47       18.0         CPE     ECCPE           ;CHECK FOR PARITY EVEN - BIT -1
2413  477F  DA  18  43       10.0         JC      TST1L           ;JUMP IF ERROR DETECTED
2414  4782  C3  AB  47       10.0         JMP     CKECC1          ;ON TO THE NEXT SECTION....
2415
2416                                        ·HERE TO CHECK FOR PARITY ODD - BIT -0
2417
2418  4785  3A  96  4A       13.0  ECCPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2419  4788  E6  10            7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2420  478A  CA  98  47       10.0         JZ      ECCPOE          ;EXIT IF OK
2421  478D                               ERRA    ECCPOE,ECCPOC
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  478D  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0015                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4790    15                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4791    00                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4792  CD  15  28       18.0                ECCPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4795  DA  98  47       10.0                JC      ECCPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2422                                        ;>PARITY BIT ON ECC SHOULD =0 AND WAS =1
2423  4798  C9               10.0  ECCPOE: RET
2424
2425                                        ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2426
2427  4799  3A  96  4A       13.0  ECCPE:  LDA     SINTSTA         ;GET THE BIT
2428  479C  E6  10            7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2429  479E  C0               12.0          RNZ
2430  479F                               ERRA    ECCPEE,ECCPEC
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  479F  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0016                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47A2    16                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47A3    00                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47A4  CD  15  28       18.0                ECCPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47A7  DA  AA  47       10.0                JC      ECCPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2431                                        ;>PARITY BIT ON ECC WAS  0 AND SHOULD BE -1
2432  47AA  C9               10.0  ECCPEE: RET
2433  47AB  CD  33  4B       18.0  CKECC1: CALL    CLECC
2434  47AE  C3  89  48       10.0          JMP     SCHREX          ;GO BACK TO CLOCK LOOP
```

```
2436  47B1                                    S
 (1)                                          ;*****************************************************************************
2437                                          ;CKACRC -- ROUTINE TO CHECK 'ACRC' CHARACTER ON WMC OUTOUT.  ALSO CHECKS
2438                                          ;       THAT THE RESIDUAL CHARACTER HAS NOT BEEN PRESENTED BEFORE 'ACRC',
2439                                          ;       THAT THE WMC HAS NOT PREVIOUSLY SET 'CRC', AND THAT A PREVIOUS
2440                                          ;       'ACRC' CHARACTER HAS NOT BEEN PRESENTED.
2441                                          ;       'HAVRCH' IS THE RESIDUAL CHARACTER FLAG.
2442                                          ;       'HAVACRC' IS THE AUX-CRC CHARACTER FLAG.
2443                                          ;       'HAVCRC' IS THE CRC CHARACTER FLAG.
2444  47B1                                    S
 (1)                                          ;*****************************************************************************
2445
2446  47B1  3A  9E  4A        13.0  CKACRC: LDA     HAVRCH          ;GET RESID FLAG
2447  47B4  A7                 4.0          ANA     A
2448  47B5  CA  BD  47        10.0          JZ      CKA1            ;JUMP IF RESIDUAL CHAR NOT RECEIVED YET
2449  47B8                                  ERR     TST1L,CKA1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  47B8  CD  09  28        18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0017                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47BB  17                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47BC  00                                 .BYTE
 (1)  47BD  CD  15  28        18.0          CKA1::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  47C0  DA  18  43        10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2450                                      ;>WMC PRESENTED RESIDUAL CHARACTER BEFORE ACRC CHARACTER
2451  47C3  3A  9C  4A        13.0          LDA     HAVCRC          ;GET CRC RECEIVED FLAG
2452  47C6  A7                 4.0          ANA     A
2453  47C7  CA  CF  47        10.0          JZ      CKA2
2454  47CA                                  ERR     TST1L,CKA2
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  47CA  CD  09  28        18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0018                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47CD  18                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47CE  00                                 .BYTE
 (1)  47CF  CD  15  28        18.0          CKA2::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  47D2  DA  18  43        10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2455                                      ;>WMC PRESENTED CRC CHARACTER BEFORE ACRC CHARACTER
2456  47D5  3A  9D  4A        13.0          LDA     HAVACRC         ;GET ACRC CHAR RECEIVED FLAG
2457  47D8  A7                 4.0          ANA     A
2458  47D9  CA  E1  47        10.0          JZ      CKA3
2459  47DC                                  ERR     TST1L,CKA3
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  47DC  CD  09  28        18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0019                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47DF  19                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47E0  00                                 .BYTE
 (1)  47E1  CD  15  28        18.0          CKA3::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  47E4  DA  18  43        10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2460                                      ;>WMC PRESENTED ACRC CHARACTER MORE THAN ONCE
2461  47E7  3A  9F  4A        13.0  CKA3A:  LDA     RPADCT          ;GET # RECEIVED PAD CHARACTERS
2462  47EA                                  ROUT    ADATA           ;SAVE ACTUAL
 (1)  47EA  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47EC  7F                 4.0                  MOV     A,A             ;RETRY LINK
2463  47ED  47                 4.0          MOV     B,A
2464  47EE  3A  A6  4A        13.0          LDA     IMPADCT         ;GET # DATA BYTES IN RESID. GROUP
```

```
2465  47F1  3D                4.0          DCR     A
2466  47F2                             CKA3B:  ROUT    EDATA
 (1)  47F2  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47F4  7F                   4.0                  MOV     A,A             ;RETRY LINK
2467  47F5  B8                   4.0          CMP     B                       ;EXPECTED=ACTUAL
2468  47F6  CA  FE  47            10.0         JZ      CKA3C                   ;JUMP IF OK
2469  47F9                               ERRB    TST1L,CKA3C,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  47F9  CD  12  28            18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001A                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47FC  1A                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47FD  01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  47FE  CD  15  28            18.0          CKA3C:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4801  DA  18  43            10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2470                            ;>WMC FAILED TO SEND PROPER # PAD CHAR. IN RESIDUAL GROUP
2471
2472  4804  3A  A3  4A            13.0  CKAC4:  LDA     RESCNT          ;GET # RECEIVED RESIDUAL DATA BYTES
2473  4807  47                     4.0          MOV     B,A
2474  4808                               ROUT    ADATA           ;SAVE ACTUAL
 (1)  4808  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  480A  7F                   4.0                  MOV     A,A             ;RETRY LINK
2475  4808  3A  A4  4A            13.0         LDA     IMRESCNT        ;GET EXPECTED #
2476  480E                               ROUT    EDATA
 (1)  480E  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4810  7F                   4.0                  MOV     A,A             ;RETRY LINK
2477  4811  B8                   4.0          CMP     B                       ;EXPECTED=ACTUAL
2478  4812  CA  1A  48            10.0         JZ      CKAC4C                  ;JUMP IF OK
2479  4815                               ERRB    TST1L,CKAC4C
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4815  CD  12  28            18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001B                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4818  1B                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4819  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  481A  CD  15  28            18.0          CKAC4C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  481D  DA  18  43            10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2480                            ;>WMC FAILED TO SEND ALL DATA BYTES IN RESIDUAL GROUP
2481  4820  CD  6D  4A            18.0  CKA5:   CALL    ACRCFI          ;FINISH THE 'ACRC' SIMULATION CHARACTER
2482  4823  3E  FE                 7.0          MVI     A,@376
2483  4825  32  9D  4A            13.0         STA     HAVACRC         ;SET THE 'ACRC' RECEIVED FLAG
2484  4828  DB  D3                 10.0         IN      WRTDAT          ;GET ACTUAL 'ACRC' CHARACTER
2485  482A  21  A0  4A            10.0         LXI     H,ACRCHR        ;POINT TO 'EXPECTED' CHAR
2486  482D  BE                     7.0          CMP     M               ;ACTUAL 'ACRC' RECEIVED OK?
2487  482E  CA  3D  48            10.0         JZ      CKA4            ;JUMP IF OK
2488  4831                               ROUT    ADATA           ;SAVE ACTUAL 'ACRC'
 (1)  4831  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4833  7F                   4.0                  MOV     A,A             ;RETRY LINK
2489  4834  7E                     7.0          MOV     A,M             ;GET EXPECTED
2490  4835                               ROUT    EDATA           ;SAVE IT
 (1)  4835  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4837  7F                   4.0                  MOV     A,A             ;RETRY LINK
```

```
2492  4838                                          ERRB    TST1L,CKA4
 (1)                                                ,FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4838  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001C                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  483B  1C                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  483C  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  483D  CD  15  28        18.0                CKA4::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4840  DA  18  43        10.0                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2493                                        ;>WMC 'ACRC' DATA BYTE NOT SAME AS CALCULATED 'ACRC' DATA BYTE
2494                                        ;HERE TO CHECK THE PARITY BIT ON THE ACRC ASSEMBLED
2495
2496  4843  DB  D3              10.0          IN      WRTDAT          ;GET THE ACRC BYTE AGAIN
2497  4845                                    ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  4845  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4847  7F                   4.0                  MOV     A,A             ;RETRY LINK
2498  4848  A7                   4.0           ANA     A               ;SET THE CONDITION BITS
2499  4849  E4  5B  48          18.0           CPO     ACRCPO          ;CHECK FOR ODD PARITY
2500  484C  DA  18  43          10.0           JC      TST1L           ;JUMP IF ERROR DETECTED
2501  484F  DB  D3              10.0           IN      WRTDAT          ;GET THE ACRC BYTE AGAIN
2502  4851  A7                   4.0           ANA     A               ;SET THE CONDITION BITS
2503  4852  EC  6F  48          18.0           CPE     ACRCPE          ;CHECK FOR PARITY EVEN - BIT -1
2504  4855  DA  18  43          10.0           JC      TST1L           ;JUMP IF ERROR DETECTED
2505  4858  C3  81  48          10.0           JMP     CACRC1          ;ON TO THE NEXT SECTION....
2506
2507                                        ;HERE TO CHECK FOR PARITY ODD - BIT -0
2508  485B  3A  96  4A          13.0  ACRCPO: LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2509  485E  E6  10               7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2510  4860  CA  6E  48          10.0          JZ      ACRCOE          ;EXIT IF OK
2511  4863                                    ERRA    ACRCOE,ACRCOC
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4863  CD  0F  28        18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       001D                                   MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4866  1D                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4867  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4868  CD  15  28        18.0          ACRCOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  486B  DA  6E  48        10.0                  JC      ACRCOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2512                                        ;>PARITY BIT ON ACRC SHOULD =0 AND WAS =1
2513  486E  C9                  10.0  ACRCOE: RET
2514
2515                                        ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2516  486F  3A  96  4A          13.0  ACRCPE: LDA     SINTSTA         ;GET THE BIT
2517  4872  E6  10               7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2518  4874  C0                  12.0          RNZ
```

```
2520  4875                                          ERRA    ACRCEE,ACRCEC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4875  CD  0F  28      18.0                            CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       001E                                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4878  1E                                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4879  00                                               .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  487A  CD  15  28      18.0                    ACRCEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  487D  DA  80  48      10.0                            JC      ACRCEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2521                                    ;>PARITY BIT ON ACRC WAS =0 AND SHOULD BE =1
2522  4880  C9                 10.0     ACRCEE: RET
2523  4881  DB  D3             10.0     CACRC1: IN      WRTDAT          ;GET THE 'ACRC' CHARACTER AGAIN
2524  4883  CD  3D  4B        18.0             CALL    GENECC          ;GO INCLUDE THE 'ACRC' CHARACTER IN THE ECC
2525  4886  CD  D8  49        18.0             CALL    GENCRC          ;ADD IT TO 'CRC' SIMULATION CHARACTER
2526  4889  06  7F             7.0     SCHREX: MVI     B,@177          ;RESET THE WATCHDOG COUNTERS
2527  488B  48                  4.0             MOV     C,B
2528  488C  C3  07  44          10.0             JMP     TST1C           ;GO BACK TO CLOCK LOOP
```

```
                                        S
                                        ; ***************************************************************************
                                        ;CKCRC -- ROUTINE TO CHECK THE WMC 'CRC' DATA BYTE.  CHECKS THAT 'ACRC'
                                        ;       CHARACTER WAS ALREADY RECEIVED, THAT THIS CRC CHARACTER IS NOT
                                        ;       THE 7TH CRC CHARACTER, AND THAT THE ACTUAL WMC 'CRC' CHARACTER
                                        ;       IS THE SAME AS THE CALCULATED 'CRC' CHARACTER.
                                        S
                                        ; ***************************************************************************

488F  3A 9E 4A      13.0  CKCRC:  LDA     HAVRCH          ;GET THE HAVE RESIDUAL CHARACTER FLAG
4892  A7             4.0          ANA     A               ;SET THE CONDITION BITS
4893  CA 9D 48      10.0          JZ      CKCRC1          ;CONTINUE - NO RESIDUAL CHAR. YET
4896  3A A9 4A      13.0          LDA     LASTCRC         ;GET THE 'LAST CRC CHARACTER' FLAG
4899  A7             4.0          ANA     A               ;SET THE CONDITION BITS
489A  C2 A8 46      10.0          JNZ     QUIT            ;ALL CHARACTERS RECEIVED - WMC SHOULD BE DONE

489D  3A 9E 4A      13.0  CKCRC1: LDA     HAVRCH          ;GET RESIDUAL CHAR. RECEIVED FLAG
48A0  A7             4.0          ANA     A
48A1  CA A9 48      10.0          JZ      CKC1            ;JUMP IF OK-NO RESIDUAL CHAR. YET
48A4                              ERR     TST1L,CKC1
                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
48A4  CD 09 28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
      001F                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
48A7  1F                              .BYTE   MSGN            ;MESSAGE NUMBER ID
48A8  00                              .BYTE
48A9  CD 15 28      18.0        CKC1::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
48AC  DA 18 43      10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                          ;>WMC SET RESIDUAL CHARACTER BEFORE CRC CHARACTERS WERE ALL SENT

48AF  3A 9D 4A      13.0          LDA     HAVACRC         ;GET ACRC CHAR. RECEIVED FLAG
48B2  A7             4.0          ANA     A
48B3  C2 BB 48      10.0          JNZ     CKC2            ;JUMP IF RECEIVED ACRC
48B6                              ERR     TST1L,CKC2
                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
48B6  CD 09 28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
      0020                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
48B9  20                              .BYTE   MSGN            ;MESSAGE NUMBER ID
48BA  00                              .BYTE
48BB  CD 15 28      18.0        CKC2::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
48BE  DA 18 43      10.0                JC      TST1L           ;LOOP ADDRFSS IF LOOP SPECIFIED
                          ;>WMC FAILED TO SEND ACRC CHARACTER BEFORE CRC CHARACTER
48C1  06 06          7.0          MVI     B,6             ;SET THE CRC LIMIT TO 6
48C3  3A A8 4A      13.0          LDA     PDCRC           ;SEE IF ANOTHER EXPECTED
48C6  A7             4.0          ANA     A
48C7  C2 CC 48      10.0          JNZ     CRC2A           ;JUMP IF 1 MORE PAD CHAR EXPECTED
48CA  3E 07          7.0          MVI     A,7             ;SET LIMIT TO 7
48CC  3A 9C 4A      13.0  CRC2A:  LDA     HAVCRC          ;GET # CRC RECEIVED
48CF  3C             4.0          INR     A               ;UPDATE THE COUNT
48D0  32 9C 4A      13.0          STA     HAVCRC          ;SAVE IT
48D3  FE 01          7.0          CPI     1               ;1ST CRC CHARACTER?
48D5  C2 DB 48      10.0          JNZ     CRC4
48D8  CD 0D 4A      18.0          CALL    CRCFIN          ;FINISH CRC CHAR SIMULATION-ONLY ON 1ST CHAR.
48DB  DB D3         10.0  CRC4:   IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
48DD                              ROUT    ADATA
```

```
 (1)  48DD  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  48DF  7F               4.0                MOV     A,A             ;RETRY LINK
2568  48E0  21  98  4A       10.0        LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
2569  48E3  BE                7.0        CMP     M               ;ACTUAL=SIMULATED?
2570  48E4  CA  F0  48        10.0        JZ      CRC5            ;JUMP IF OK
2571  48E7  7E                7.0        MOV     A,M             ;GET SIMULATED CRC CHAR.
2572  48E8                               ROUT    EDATA
 (1)  48E8  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  48EA  7F               4.0                MOV     A,A             ;RETRY LINK
2573  48EB                               ERRB    TST1L,CRC5
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  48EB  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0021                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48EE  21                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48EF  00                               .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48F0  CD  15  28       18.0        CRC5::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48F3  DA  18  43       10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2574                                   ;>WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
2575
2576                                   ;HERE TO CHECK THE PARITY BIT ON THE CRC ASSEMBLED
2577
2578  48F6  DB  D3               10.0        IN      WRTDAT          ;GET THE CRC BYTE AGAIN
2579  48F8                               ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  48F8  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  48FA  7F               4.0                MOV     A,A             ;RETRY LINK
2580  48FB  A7                4.0        ANA     A               ;SET THE CONDITION BITS
2581  48FC  E4  0E  49        18.0        CPO     CRCPO           ;CHECK FOR ODD PARITY
2582  48FF  DA  18  43        10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2583  4902  DB  D3            10.0        IN      WRTDAT          ;GET THE CRC BYTE AGAIN
2584  4904  A7                4.0        ANA     A               ;SET THE CONDITION BITS
2585  4905  EC  22  49        18.0        CPE     CRCPE           ;CHECK FOR PARITY EVEN - BIT -1
2586  4908  DA  18  43        10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2587  490B  C3  34  49        10.0        JMP     CKCRC2          ;ON TO THE NEXT SECTION....
2588
2589                                   ;HERE TO CHECK FOR PARITY ODD - BIT =0
2590
2591  490E  3A  96  4A        13.0  CRCPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2592  4911  E6  10             7.0        ANI     WDR.P           ;SAVE THE PARITY BIT
2593  4913  CA  21  49        10.0        JZ      CRCPOE          ;EXIT IF OK
2594  4916                               ERRA    CRCPOE,CRCPOC
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4916  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0022                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4919  22                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  491A  00                               .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  491B  CD  15  28       18.0        CRCPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  491E  DA  21  49       10.0                JC      CRCPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2595                                   ;>PARITY BIT ON CRC SHOULD =0 AND WAS =1
2596  4921  C9                10.0  CRCPOE: RET
2597                                   ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2598  4922  3A  96  4A        13.0  CRCPE:  LDA     SINTSTA         ;GET THE BIT
2599  4925  E6  10             7.0        ANI     WDR.P           ;SAVE THE PARITY BIT
2600  4927  C0                12.0        RNZ
```

```
                                                ERRA    CRCPEE,CRCPEC
                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
                                                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                CRCPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      CRCPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PARITY BIT ON CRC WAS =0 AND SHOULD BE =1
                                CRCPEE: RET
                                CKCRC2: IN      WRTDAT          ;GET THE CRC CHARACTER AGAIN
                                        CALL    GENECC          ;GO INCLUDE THE PAD CHARACTER IN THE ECC
                                        JMP     SCHREX          ;BACK TO CLOCK LOOP
```

```
2609  493C                                  S
 (1)                                        ; ***************************************************************************
2610                                        ;CKRESID -- ROUTINE TO CHECK THE WMC RESIDUAL CHARACTER AND RESIDUAL
2611                                        ;       DATA GROUP BYTE RECEIVED COUNT, THAT 'ACRC' WAS RECEIVED, THAT
2612                                        ;       6 'CRC' CHARACTERS WERE RECEIVED, AND THAT THE PROPER AMOUNT
2613                                        ;       OF PAD CHARACTERS WERE RECEIVED.
2614  493C                                  S
 (1)                                        ; ***************************************************************************
2615
2616  493C  3A  9D  4A      13.0  CKRESID: LDA     HAVACRC          ;GET ACRC CHAR RECEIVED FLAG
2617  493F  A7                4.0          ANA     A
2618  4940  C2  48  49       10.0          JNZ     CKR1             ;JUMP IF ACRC RECEIVED-OK
2619  4943                                 ERR     TST1L,CKR1
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4943  CD  09  28        18.0                    CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0024                          MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4946  24                            .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4947  00                            .BYTE
 (1)  4948  CD  15  28        18.0        CKR1::  CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  494B  DA  18  43        10.0                JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
2620                                    ;>WMC FAILED TO SEND THE 'ACRC' CHARACTER BEFORE RESIDUAL CHARACTER WAS PRESENTED
2621
2622  494E  06  06                7.0          MVI     B,6              ;GET # EXPECTED CRC CHAR
2623  4950  3A  A8  4A       13.0          LDA     PDCRC            ;SEE IF 1 LESS
2624  4953  A7                4.0          ANA     A                ;IF PADCRC IS SET, MAKE COUNT 5
2625  4954  CA  59  49       10.0          JZ      CKR1A
2626  4957  06  05                7.0          MVI     B,5
2627  4959  3A  9C  4A       13.0  CKR1A:  LDA     HAVCRC           ;GET CRC''CHAR RECEIVED COUNT
2628  495C  B8                4.0          CMP     B                ;HAVE PROPER AMT OF CRC BYTES?
2629  495D  CA  6C  49       10.0          JZ      CKR2             ;JUMP IF OK
2630  4960                                 ROUT    ADATA            ;SAVE ACTUAL # OF RECEIVED CRC CHAR.
 (1)  4960  D3  94            10.0                    OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4962  7F                4.0                     MOV     A,A              ;RETRY LINK
2631  4963  78                4.0          MOV     A,B              ;GET THE EXPECTED COUNT
2632  4964                                 ROUT    EDATA            ;SAVE FOR ERROR PRINTOUT
 (1)  4964  D3  95            10.0                    OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  4966  7F                4.0                     MOV     A,A              ;RETRY LINK
2633  4967                                 ERRB    TST1L,CKR2
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4967  CD  12  28        18.0                    CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0025                          MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  496A  25                            .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  496B  00                            .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  496C  CD  15  28        18.0        CKR2::  CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  496F  DA  18  43        10.0                JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
2634                                    ;>WMC FAILED TO PRESENT CORRECT # CRC CHARACTERS IN CRC GROUP - ACTUAL = # RECEIVED
2635  4972  3A  9B  4A       13.0          LDA     RCHRIM           ;GET EXPECTED RESIDUAL CHAR
2636  4975                                 ROUT    EDATA            ;SAVE EXPECTED
 (1)  4975  D3  95            10.0                    OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  4977  7F                4.0                     MOV     A,A              ;RETRY LINK
2637  4978  47                4.0          MOV     B,A
2638  4979  DB  D3                10.0          IN      WRTDAT           ;GET ACTUAL 'RESID' CHAR.
2639  497B  B8                4.0          CMP     B                ;ACTUAL-EXPECTED
2640  497C  CA  87  49       10.0          JZ      CKR5             ;JUMP IF OK
```

```
2641  497F                                    ROUT    ADATA
 (1)  497F  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4981  7F               4.0                     MOV     A,A             ;RETRY LINK
2642  4982                                    ERRB    TST1L,CKR5
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4982  CD  12  28      18.0                             CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0026                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4985  26                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4986  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4987  CD  15  28      18.0                     CKR5::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  498A  DA  18  43      10.0                             JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2643                                    ;>WMC ACTUAL RESIDUAL DATA BYTE PRESENTED NOT SAME AS EXPECTED RESIDUAL CHAR
2644
2645                                    ;HERE TO CHECK THE PARITY BIT ON THE RESIDUAL ASSEMBLED
2646
2647  498D  DB  D3               10.0             IN      WRTDAT          ;GET THE RESIDUAL BYTE AGAIN
2648  498F                                       ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  498F  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4991  7F               4.0                     MOV     A,A             ;RETRY LINK
2649  4992  A7               4.0             ANA     A               ;SET THE CONDITION BITS
2650  4993  E4  A5  49       18.0             CPO     RESPO           ;CHECK FOR ODD PARITY
2651  4996  DA  18  43       10.0             JC      TST1L           ;JUMP IF ERROR DETECTED
2652  4999  DB  D3           10.0             IN      WRTDAT          ;GET THE RESIDUAL BYTE AGAIN
2653  499B  A7               4.0             ANA     A               ;SET THE CONDITION BITS
2654  499C  EC  B9  49       18.0             CPE     RESPE           ;CHECK FOR PARITY EVEN - BIT =1
2655  499F  DA  18  43       10.0             JC      TST1L           ;JUMP IF ERROR DETECTED
2656  49A2  C3  CB  49       10.0             JMP     CKRES1          ;ON TO THE NEXT SECTION....
2657
2658                                    ;HERE TO CHECK FOR PARITY ODD - BIT =0
2659
2660  49A5  3A  96  4A        13.0     RESPO: LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2661  49A8  E6  10             7.0             ANI     WDR.P           ;SAVE THE PARITY BIT
2662  49AA  CA  B8  49        10.0             JZ      RESPOE          ;EXIT IF OK
2663  49AD                                     ERRA    RESPOE,RESPOC
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  49AD  CD  0F  28      18.0                             CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0027                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49B0  27                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49B1  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  49B2  CD  15  28      18.0                     RESPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49B5  DA  B8  49      10.0                             JC      RESPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2664                                    ;>PARITY BIT ON RESIDUAL SHOULD -0 AND WAS -1
2665  49B8  C9                10.0     RESPOE: RET
2666
2667                                    ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2668
2669  49B9  3A  96  4A        13.0     RESPE: LDA     SINTSTA         ;GET THE BIT
2670  49BC  E6  10             7.0             ANI     WDR.P           ;SAVE THE PARITY BIT
2671  49BE  C0                12.0             RNZ
```

```
2673  49BF                                          ERRA    RESPEE,RESPEC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  49BF  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0028                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49C2  28                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49C3  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  49C4  CD  15  28        18.0                  RESPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49C7  DA  CA  49        10.0                          JC      RESPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2674                                    ;>PARITY BIT ON RESIDUAL WAS =0 AND SHOULD BE =1
2675  49CA  C9                 10.0  RESPEE: RET
2676  49CB  DB  D3             10.0  CKRES1: IN      WRTDAT          ;GET THE RESIDUAL CHARACTER AGAIN
2677  49CD  CD  3D  4B         18.0          CALL    GENECC          ;GO INCLUDE THE RESIDUAL CHARACTER IN THE ECC
2678  49D0  3E  FF              7.0          MVI     A,@377
2679  49D2  32  9E  4A         13.0          STA     HAVRCH          ;SET HAVE RESID CHAR FLAG
2680  49D5  C3  89  48         10.0          JMP     SCHREX          ;BACK TO THE CLOCK LOOP
```

```
2682  49D8                                  S
 (1)                                        ;*****************************************************************************
2683                                        ;CRC -- ROUTINE TO SIMULATE THE CRC GENERATOR IN THE TM78.  CALLED
2684                                        ;       WITH ACCUMULATOR CONTAINING DATA BYTE TO BE ADDED TO CRCWORD
2685  49D8                                  S
 (1)                                        ;*****************************************************************************
2686
2687  49D8  CD  76  4A        18.0  GENCRC: CALL    PARITY          ;CALC. PARITY ON DATA IN ACCUMULATOR
2688  49DB  F5                12.0        PUSH    PSW
2689  49DC  C5                12.0        PUSH    B
2690  49DD  E5                12.0        PUSH    H
2691  49DE  21  98  4A        10.0        LXI     H,CRCCHR        ;POINT TO CRC CHAR STORAGE
2692  49E1  AE                 7.0        XRA     M               ;XOR DATA IN ACCUMULATOR WITH PREVIOUS CRC 7 TO 0
2693  49E2  23                 6.0        INX     H               ;POINT TO CRC PARITY BIT STORAGE
2694  49E3  47                 4.0        MOV     B,A             ;SAVE RESULT OF XOR
2695  49E4  3A  A2  4A        13.0        LDA     PBIT            ;GET PARITY OF DATA BYTE ON CALL
2696  49E7  AE                 7.0        XRA     M               ;XOR WITH CRC PARITY BIT
2697  49E8  1F                 4.0        RAR                     ;PUT PARITY BIT IN 'C' BIT POSITION
2698  49E9  78                 4.0        MOV     A,B             ;GET XOR OF 7 TO 0
2699  49EA  1F                 4.0        RAR                     ;SHIFT RIGHT
2700  49EB  D2  F9  49        10.0        JNC     CRC2            ;COMPLIMENT BITS 5,4,3,+2 IF 'C' IS SET
2701
2702  49EE  EE  3C             7.0  CRC1:  XRI     @74             ;COMPLIMENT BITS
2703  49F0  32  98  4A        13.0        STA     CRCCHR          ;SAVE THE BYTE
2704  49F3  3E  01             7.0        MVI     A,1             ;AND SET THE 'P' BIT TO A 1 FOR NEXT TIME
2705  49F5  77                 7.0        MOV     M,A             ;SAVE THE 1
2706  49F6  C3  FE  49        10.0        JMP     CRC3            ;EXIT
2707
2708  49F9  32  98  4A        13.0  CRC2:  STA     CRCCHR          ;SAVE CRC CHARACTER
2709  49FC  AF                 4.0        XRA     A               ;CLEAR THE 'P' BIT
2710  49FD  77                 7.0        MOV     M,A             ;FOR THE NEXT TIME
2711  49FE  E1                10.0  CRC3:  POP     H
2712  49FF  C1                10.0        POP     B
2713  4A00  F1                10.0        POP     PSW
2714  4A01  C9                10.0        RET
2715
2716                                        ;INITIALIZE CRC CHARACTER STORAGE
2717
2718  4A02  AF                 4.0  CLCRC:  XRA     A
2719  4A03  32  98  4A        13.0        STA     CRCCHR
2720  4A06  32  9A  4A        13.0        STA     CRCCNT
2721  4A09  32  99  4A        13.0        STA     CRCCHR+1
2722  4A0C  C9                10.0        RET
2723
2724                                        ;TRANSFER COMPLETE... FINISH THE CRC GENERATION
2725
2726  4A0D  3A  98  4A        13.0  CRCFIN: LDA     CRCCHR          ;GET CRC WORD
2727  4A10  EE  D7             7.0        XRI     @327            ;COMPLIMENT BITS 7,6,4,2,1,+0
2728  4A12  32  98  4A        13.0        STA     CRCCHR
2729  4A15  C9                10.0        RET
```

```
                                        .SBTTL  CLEAR ACRC SUBROUTINE

                                        S
                                        ; ***************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ACRC CHARACTER AND ACRC CHARACTER PARITY -
                                        ;MEMORY LOCATIONS "ACRCHR" AND "ACRCP" - USED BY THE ACRC SUBROUTINE.
                                        ;IT SHOULD BE CALLED PRIOR TO STARTING A DATA TRANSFER.
                                        S
                                        ; ***************************************************************************

                                4.0     CLACRC: XRA     A               ;CLEAR THE ACCUMULATOR
                                13.0            STA     ACRCHR          ;CLEAR ACRC CHARACTER
                                13.0            STA     ACRCP           ;CLEAR ACRC PARITY
                                10.0            RET

                                        .SBTTL  CALCULATE ACRC CHARACTER

                                        S
                                        ; ***************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND THE
                                        ;CONTENTS OF "ACRCHR" AND "ACRCP" TO UPDATE THE CONTENTS OF "ACRCHR"
                                        ;AND "ACRCP" ACCORDING TO THE ANSI STANDARD POLYNOMIAL.
                                        S
                                        ; ***************************************************************************

                                12.0    GENACRC: PUSH   PSW             ;SAVE THE DATA BYTE IN REG A
                                12.0             PUSH   B               ;SAVE REGISTERS B&C
                                12.0             PUSH   D               ;SAVE REGISTERS D&E
                                12.0             PUSH   H               ;SAVE REGISTERS H&L
                                4.0              MOV    E,A             ;SAVE THE CHARACTER
                                18.0             CALL   PARITY          ;GENERATE ODD PARITY FOR CHARACTER
                                13.0             LDA    ACRCHR          ;GET THE PREVIOUS ACRC CHARACTER
                                4.0              XRA    E               ;XOR DATA AND PREVIOUS ACRC CHARACTER
                                4.0              MOV    E,A             ;SAVE XOR RESULT IN REG E

                                13.0             LDA    PBIT            ;GET THE CHARACTER PARITY BIT
                                4.0              MOV    D,A             ;SAVE IN REG D
                                13.0             LDA    ACRCP           ;GET THE ACRC PARITY BIT
                                4.0              XRA    D               ;XOR THE DATA PARITY BIT AND PREVIOUS
                                                                        ;ACRC PARITY BIT
                                4.0              MOV    D,A             ;SAVE XOR RESULT IN REG D
                                                                        ;
                                4.0              MOV    A,E             ;GET THE BYTE XOR FROM REG E
                                7.0              ANI    $08             ;BIT 3 SET?
                                4.0              MOV    A,E             ;GET THE BYTE XOR FROM REG E
                                10.0             JZ     ACRC1           ;NO-LEAVE RESULT ALONE
                                7.0              XRI    @140            ;ELSE-XOR RESULT WITH 140 - BITS 5 & 6

                                4.0     ACRC1:   MOV    E,A             ;STORE RESULT BACK IN REG. E
                                4.0              XRA    A               ;CLEAR A
                                4.0              MOV    C,A             ;CLEAR THE TRANSLATE RESULT
                                                                        ;
                                10.0             LXI    H,ACRCTB        ;POINT TO ACRC TRANSLATION TBL
                                18.0             CALL   TRANS           ;TRANSLATE THE BITS
```

```
2781                                                    ;
2782  4A46  7B              4.0          MCV     A,E     ;GET THE BYTE BEING TRANSLATED
2783  4A47  E6  02           7.0          ANI     $02     ;BIT 1 SET
2784  4A49  3E  00           7.0          MVI     A,$00   ;CLEAR A
2785  4A4B  CA  50  4A       10.0         JZ      ACRC2   ;GO CLEAR ACRC P BIT IF ZERO
2786  4A4E  3E  01           7.0          MVI     A,$01   ;ELSE-SET ACRC P BIT
2787  4A50  32  A1  4A       13.0 ACRC2:  STA     ACRCP   ;MODIFY ACRC P BIT
2788                                                    ;
2789  4A53  7A               4.0          MOV     A,D     ;GET THE P BIT XOR RESULT
2790  4A54  A7               4.0          ANA     A       ;SET THE CONDITION CODE
2791  4A55  CA  5C  4A       10.0         JZ      ACRC3   ;CONTINUE IF ZERO
2792  4A58  79               4.0          MOV     A,C     ;ELSE-GET TRANSLATION RESULTS
2793
2794  4A59  F6  01           7.0          ORI     $01     ;OR IN TRANSLATION BIT
2795  4A5B  4F               4.0          MOV     C,A     ;SAVE RESULTS IN C
2796  4A5C  79               4.0 ACRC3:   MOV     A,C     ;GET THE TRANSLATED BYTE
2797  4A5D  32  A0  4A       13.0         STA     ACRCHR  ;STORE IN ACRC CHARACTER
2798  4A60  E1               10.0         POP     H       ;RESTORE REGISTERS H&L
2799  4A61  D1               10.0         POP     D       ;RESTORE REGISTERS D&E
2800  4A62  C1               10.0         POP     B       ;RESTORE REGISTERS B&C
2801  4A63  F1               10.0         POP     PSW     ;RESET THE ORRIG. DATA BYTE IN REG A
2802  4A64  C9               10.0         RET             ;RETURN
2803
2804  4A65  40                     ACRCTB: $40             ;BIT 0 - POSITION 6
2805  4A66  00                             $00             ;BIT 1 = NO REPLACEMENT
2806  4A67  20                             $20             ;BIT 2 = POSITION 5
2807  4A68  04                             $04             ;BIT 3 = POSITION 2
2808  4A69  80                             $80             ;BIT 4 = POSITION 7
2809  4A6A  02                             $02             ;BIT 5 = POSITION 1
2810  4A6B  10                             $10             ;BIT 6 = POSITION 4
2811  4A6C  08                             $08             ;BIT 7   POSITION 3
```

```
                                        .SBTTL  FORMAT FINAL ACRC CHARACTER

                                        S
                                        ; ***************************************************************************
 (1)                                    ;THIS SUBROUTINE TAKES THE CONTENTS OF "ACRCHR" AND "ACRCP" AND
                                        ;TRANSFORMS THEM INTO A 8 BIT ACRC CHARACTER.  THE FINAL ACRC CHARACTER
                                        ;IS AVAILABLE IN "ACRCHR".
                                        S
                                        ; ***************************************************************************
 (1)
4A6D 3A A0 4A          13.0 ACRCFI: LDA     ACRCHR             ;GET THE ACRC CHAR
4A70 EE BC              7.0          XRI     @274               ;INVERT THE PROPER BITS
4A72 32 A0 4A          13.0          STA     ACRCHR             ;STORE THE FINAL CHARACTER
4A75 C9                10.0          RET                        ;RETURN

                                        .SBTTL  PARITY CALCULATION ROUTINE

                                        ;THIS SUBROUTINE CALCULATES ODD PARITY ON THE CONTENTS OF THE
                                        ;ACCUMULATOR AND SETS OR CLEARS THE MEMORY LOCATION 'PBIT AS REQUIRED.

4A76 F5                12.0 PARITY: PUSH    PSW                ;SAVE ACCUMULATOR
4A77 A7                 4.0          ANA     A                  ;SET CONDITION BITS
4A78 3E 00              7.0          MVI     A,$00              ;CLEAR ACCUMULATOR
4A7A E2 7F 4A          10.0          JPO     PAR1               ;SKIP IF PARITY ODD
4A7D 3E 01              7.0          MVI     A,$01              ;ELSE-LOAD ACC. WITH 1
4A7F 32 A2 4A          13.0 PAR1:   STA     PBIT               ;SET UP THE P BIT
4A82 F1                10.0          POP     PSW                ;RESTORE ACCUMULATOR
4A83 C9                10.0          RET                        ;RETURN

                                        ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE

4A84 00                         CCTLWD: .BYTE   0               ;IMAGE OF CLOCK CONTROL WORD
4A85 00                         IMBYTC: .BYTE   0               ;IMAGE OF BYTE COUNTER 7-0 BITS

4A86 00                         ABCNT:  .BYTE   0               ;ASSEMBLED BYTE COUNTER
4A87 00                         ABCNTR: .BYTE   0
4A88 00                         EABCNT: .BYTE   0               ;BYTE COUNT AT A DETECTED ERROR

4A89 00                         FORMT:  .BYTE   0               ;FORMAT CODE SELECTED FOR TEST
4A8A 00                         DPNUM:  .BYTE   0               ;DATA PATTERN OFFSET NUMBER (0 TO 17)
4A8B 00                         CLKON:  .BYTE   0               ;'SCLK' ON FLAG-USED TO TELL WHEN TO FILL DDR REGS
4A8C 00                         ECCCNT: .BYTE   0               ;COUNT BY 7 FOR ECC CHARACTER CHECK
4A8D 00                         CLKCNT: .BYTE   0               ;COUNT OF SCLKS GENERATED

4A8E 00                         SDDRA:  .BYTE   0               ;DDRA DATA ACTUALLY LOADED
4A8F 00                         SDDRB:  .BYTE   0               ;DDRB DATA ACTUALLY LOADED
4A90 00                         SDDRC:  .BYTE   0               ;DDRC DATA ACTUALLY LOADED

4A91 00                         SDDRAT: .BYTE   0               ;DDRA DATA FROM PGM TABLE
4A92 00                         SDDRBT: .BYTE   0               ;DDRB DATA FROM PGM TABLE
4A93 00                         SDDRCT: .BYTE   0               ;DDRC DATA FROM PGM TABLE

4A94 00                         SWMCSTA: .BYTE  0               ;COPY OF THE WMC STATUS AT 'XFER' BIT TIME
4A95 00                         SXFER:  .BYTE   0               ;SOFTWARE 'XFER' FLAG
```

```
2865  4A96  00                SINTSTA: .BYTE  0           ;SOFTWARE COPY OF INTSTA WORD
2866
2867  4A97  00                ECCCHR: .BYTE   0           ;ECC CHARACTER CALCULATED BY THE SOFTWARE
2868  4A98  00  00            CRCCHR: .WORD   0           ;IMAGE OF THE CRC CHARACTER
2869  4A9A  00                CRCCNT: .BYTE   0           ;# CRC CHAR RECEIVED
2870  4A9B  00                RCHRIM: .BYTE   0           ;IMAGE OF 'RESIDUAL' CHAR. TO BE SEEN ON WMC WHEN 'RESID
2871  4A9C  00                HAVCRC: .BYTE   0           ;RECEIVED 'CRC' CHAR
2872  4A9D  00                HAVACRC: .BYTE  0           ;RECEIVED 'ACRC' CHAR
2873  4A9E  00                HAVRCH: .BYTE   0           ;FLAG TO INDICATE 'RESID CHAR' RECEIVED
2874  4A9F  00                RPADCT: .BYTE   0           ;# RECEIVED PAD CHAR
2875  4AA0  00                ACRCHR: .BYTE   0           ;ACRC CHARACTER
2876  4AA1  00                ACRCP:  .BYTE   0           ;ACRC CHARACTER PARITY
2877  4AA2  00                PBIT:   .BYTE   0
2878  4AA3  00                RESCNT: .BYTE   0           ;RESIDUAL CHARACTERS COUNT
2879  4AA4  00                IMRESCNT: .BYTE 0           ;SOFTWARE COPY OF RESIDUAL COUNT GIVEN TO THE WMC
2880  4AA5  00                CLKCRC: .BYTE   0           ;EXTRA CLOCK-TO-CRC FLAG
2881  4AA6  00                IMPADCT: .BYTE  0           ;SOFTWARE COPY OF THE PAD COUNT GIVEN TO THE WMC
2882  4AA7  00                TPBCNT: .BYTE   0           ;TEST PASS BYTE COUNTER
2883  4AA8  00                PDCRC:  .BYTE   0           ;PAD/CRC FLAG IF SET, WMC WILL ISSUE A PAD
2884  4AA9  00                LASTCRC: .BYTE  0           ;LAST CRC BYTE EXPECTED FLAG
```

```
2886  4AAA                                  S
 (1)                                        ; ***************************************************************************
2887                                        ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT ''SCLK'' CYCLE.
2888                                        ;       DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
2889                                        ;       'SDDRCC' FOR REFERENCE.
2890  4AAA                                  S
 (1)                                        ; ***************************************************************************
2891
2892  4AAA  F5              12.0  GENDAT: PUSH    PSW             ;SAVE PSW + A
2893  4AAB  C5              12.0          PUSH    B               ;SAVE B + C
2894  4AAC  E5              12.0          PUSH    H               ;SAVE H + L
2895  4AAD  3A  8A  4A      13.0          LDA     DPNUM           ;GET THE LAST PATTERN # USED
2896  4AB0  3C               4.0          INR     A               ;POINT TO NEXT
2897  4AB1  32  8A  4A      13.0          STA     DPNUM           ;SAVE NUMBER
2898  4AB4  4F               4.0          MOV     C,A
2899  4AB5  06  00           7.0          MVI     B,0             ;B + C HAVE OFFSET
2900  4AB7  21  EF  4A      10.0          LXI     H,CDATBL        ;POINT TO DATA TABLE
2901  4ABA  09              10.0          DAD     B               ;ADD IN OFFSET TO H + L
2902
2903  4ABB  7E               7.0          MOV     A,M             ;GET DATA BYTE FOR DDR 'A'
2904  4ABC  32  91  4A      13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
2905  4ABF  AF               4.0          XRA     A               ;
2906  4AC0  32  92  4A      13.0          STA     SDDRBT          ;SAVE DDR 'B' IMAGE
2907  4AC3  32  93  4A      13.0          STA     SDDRCT          ;SAVE DDR 'C' IMAGE
2908
2909  4AC6  E1              10.0  GENOUT: POP     H               ;RESTORE REGS
2910  4AC7  C1              10.0          POP     B
2911  4AC8  F1              10.0          POP     PSW
2912  4AC9  C9              10.0          RET
```

```
                                        S
                                        ; ***************************************************************************
                                        ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
                                        ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
                                        ;       THE MASSBUS. WMC 'LEFT' FLAG SET INDICATES THAT THIS ROUTINE IS NOT
                                        ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
                                        ;       'LEFT' FLAG IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
                                        ;       DDR FORMAT.
                                        S
                                        ; ***************************************************************************

                                        ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR INJECTION

4ACA 3A 91 4A      13.0 SETDDR: LDA     SDDRAT          ;GET DDRA TEMP
4ACD E6 3F          7.0         ANI     @77             ;STRIP 2 BITS OFF
4ACF 07             4.0         RLC
4AD0 07             4.0         RLC                     ;LEFT JUSTIFY
4AD1 47             4.0         MOV     B,A             ;TEMP SAVE
4AD2 3A 93 4A      13.0         LDA     SDDRCT          ;GET DDRC TEMP
4AD5 E6 03          7.0         ANI     @3              ;SAVE 2 BITS
4AD7 B0             4.0         ORA     B
4AD8 32 8E 4A      13.0         STA     SDDRA           ;SAVE DDRA DESIRED
4ADB D3 D8         10.0         OUT     DDRA            ;LOAD THE REAL DDRA

4ADD 3A 91 4A      13.0         LDA     SDDRAT          ;GET DDRA AGAIN
4AE0 E6 C0          7.0         ANI     @300            ;SAVE 2 BITS
4AE2 07             4.0         RLC
4AE3 07             4.0         RLC
4AE4 32 8F 4A      13.0         STA     SDDRB           ;SAVE DDRB DATA
4AE7 D3 D9         10.0         OUT     DDRB            ;LOAD THE REAL DDRB

4AE9 3A 90 4A      13.0         LDA     SDDRC
4AEC D3 DA         10.0         OUT     DDRC
4AEE C9            10.0         RET                     ;EXIT
```

```
2947  4AEF  F7                    CDATBL: .BYTE   @367            ;IST SET OF DATA HAS KNOWN RESULTS
2948  4AF0  FB                            .BYTE   @373
2949  4AF1  00                            .BYTE   @0
2950  4AF2  23                            .BYTE   @43
2951  4AF3  88                            .BYTE   @210
2952  4AF4  C0                            .BYTE   @300
2953  4AF5  63                            .BYTE   @143            ;EXPECTED 'ECC' IS @311
2954                                                               ;EXPECTED INTERMEDIATE 'CRC' IS @142
2955                                                               ;EXPECTED INTERMEDIATE 'ACRC' IS @144
2956
2957  4AF6  12                            .BYTE   $12             ;HEX 12
2958  4AF7  23                            .BYTE   $23             ;HEX 23
2959  4AF8  34                            .BYTE   $34             ;HEX 34
2960  4AF9  45                            .BYTE   $45             ;HEX 45
2961  4AFA  56                            .BYTE   $56             ;HEX 56
2962  4AFB  67                            .BYTE   $67             ;HEX 67
2963  4AFC  78                            .BYTE   $78             ;HEX 78
2964  4AFD  89                            .BYTE   $89             ;HEX 89
2965  4AFE  9A                            .BYTE   $9A             ;HEX 9A
2966  4AFF  AB                            .BYTE   $AB             ;HEX AB
2967  4B00  BC                            .BYTE   $BC             ;HEX BC
2968  4B01  CD                            .BYTE   $CD             ;HEX CD
2969  4B02  DE                            .BYTE   $DE             ;HEX DE
2970  4B03  EF                            .BYTE   $EF             ;HEX EF
2971  4B04  F0                            .BYTE   $F0             ;HEX F0
2972  4B05  01                            .BYTE   $01             ;HEX 01
2973  4B06  02                            .BYTE   $02             ;FLOATING 1'S
2974  4B07  04                            .BYTE   $04
2975  4B08  08                            .BYTE   $08
2976  4B09  10                            .BYTE   $10
2977  4B0A  20                            .BYTE   $20
2978  4B0B  40                            .BYTE   $40
2979  4B0C  80                            .BYTE   $80
2980  4B0D  FE                            .BYTE   $FE             ;FLOATING 0'S
2981  4B0E  FD                            .BYTE   $FD
2982  4B0F  FB                            .BYTE   $FB
2983  4B10  F7                            .BYTE   $F7
2984  4B11  EF                            .BYTE   $EF
2985  4B12  DF                            .BYTE   $DF
2986  4B13  BF                            .BYTE   $BF
2987  4B14  7F                            .BYTE   $7F
2988  4B15                        DTEND:                          ;END OF THE DATA TABLE
2989
2990                                      ;ROUTINE TO SHIFT CONTENTS OF ACCUMULATOR RIGHT OR LEFT EITHER 2 OR 4 TIMES
2991
2992  4B15  E6  FC          7.0   SHR4:   ANI     @374            ;STRIP OFF 2 BITS
2993  4B17  0F              4.0           RRC                     ;SHIFT RIGHT 2 TIMES
2994  4B18  0F              4.0           RRC
2995  4B19  E6  FC          7.0   SHR2:   ANI     @374            ;STRIP OFF 2 BITS
2996  4B1B  0F              4.0           RRC                     ;SHIFT RIGHT 2 TIMES
2997  4B1C  0F              4.0           RRC
2998  4B1D  C9             10.0           RET
2999
3000  4B1E  E6  3F          7.0   SHL4:   ANI     @77             ;SHIFT LEFT TWICE
```

```
3001  4B20  07                4.0          RLC
3002  4B21  07                4.0          RLC
3003  4B22  E6  3F            7.0  SHL2:   ANI     @77
3004  4B24  07                4.0          RLC                    ;SHIFT LEFT TWICE
3005  4B25  07                4.0          RLC
3006  4B26  C9               10.0          RET
3007
3008
3009  4B27                          S
 (1)                              ; *********************************************************************
3010                              ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
3011                              ;       'CLK' BIT TO A '1' THEN A '0'.
3012  4B27                        S
 (1)                              ; *********************************************************************
3013
3014  4B27  3A  84  4A        13.0  CLKSYS: LDA     CCTLWD           ;GET SOFTWARE CLOCK CONTROL IMAGE
3015  4B2A  F6  40             7.0          ORI     SSCLK            ;ADD IN 'CLK' BIT
3016  4B2C  D3  F0            10.0          OUT     CLKCTL           ;LOAD CLOCK CONTROL
3017  4B2E  E6  3F             7.0          ANI     @77              ;STRIP OFF CLOCK BIT
3018  4B30  D3  F0            10.0          OUT     CLKCTL           ;LOAD CLOCK CONTROL WORD
3019  4B32  C9                10.0          RET                      ;EXIT - CLOCK CYCLE COMPLETE
```

```
                                        .SBTTL  CLEAR ECC SUBROUTINE
4B33                                    S
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
                                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
4B33                                    S
 (1)                                    ; ************************************************************************

4B33  F5                    12.0  CLECC:  PUSH    PSW             ;SAVE THE ACCUMULATOR
4B34  AF                     4.0          XRA     A               ;CLEAR THE ACCUMULATOR
4B35  32  97  4A            13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
4B38  32  8C  4A            13.0          STA     ECCCNT          ;CLEAR THE COUNT OF ASSEMBLED BYTES
4B3B  F1                    10.0          POP     PSW             ;RESTORE THE ACCUMULATOR
4B3C  C9                    10.0          RET                     ;RETURN TO USER

                                        .SBTTL  CALCUALTE ECC CHARACTER
4B3D                                    S
 (1)                                    ; ************************************************************************
                                        ;GENECC --THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF 'ECCCHR' TO UPDATE THE CONTENTS OF 'ECCCHR' ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
4B3D                                    S
 (1)                                    ; ************************************************************************

4B3D  F5                    12.0  GENECC: PUSH    PSW             ;SAVE THE ACCUMULATOR
4B3E  C5                    12.0          PUSH    B               ;SAVE B&C
4B3F  D5                    12.0          PUSH    D               ;SAVE D&E
4B40  E5                    12.0          PUSH    H               ;SAVE H&L
4B41  21  97  4A            10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
4B44  AE                     7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
4B45  5F                     4.0          MOV     E,A             ;SAVE XOR RESULT IN E
4B46  E6  10                 7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
4B48  7B                     4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
4B49  CA  4E  4B            10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
4B4C  EE  23                 7.0          XRI     $23             ;ELSE-XOR WITH 23
4B4E  5F                     4.0  ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
4B4F  AF                     4.0          XRA     A               ;CLEAR A
4B50  4F                     4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
4B51  21  60  4B            10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
4B54  CD  68  4B            18.0          CALL    TRANS           ;TRANSLATE THE BITS
4B57  79                     4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
4B58  32  97  4A            13.0          STA     ECCCHR          ;STORE RESULT
4B5B  E1                    10.0          POP     H               ;RESTORE H&L
4B5C  D1                    10.0          POP     D               ;RESTORE D&E
4B5D  C1                    10.0          POP     B               ;RESTORE B&C
4B5E  F1                    10.0          POP     PSW             ;RESTORE ACCUM.
4B5F  C9                    10.0          RET
```

```
3066  4B60  08                        ECCTBL: $08                     ;BIT 0 = POSITION 3
3067  4B61  20                                $20                     ;BIT 1 = POSITION 5
3068  4B62  02                                $02                     ;BIT 2 = POSITION 1
3069  4B63  40                                $40                     ;BIT 3 = POSITION 6
3070  4B64  80                                $80                     ;BIT 4 = POSITION 7
3071  4B65  01                                $01                     ;BIT 5 = POSITION 0
3072  4B66  10                                $10                     ;BIT 6 = POSITION 4
3073  4B67  04                                $04                     ;BIT 7 = POSITION 2
3074
3075
3076                                          .SBTTL  POLYNOMIAL BIT TRANSLATION
3077
3078                                    ;
3079                                    ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
3080                                    ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
3081                                    ;
3082
3083  4B68  06  01            7.0   TRANS:  MVI     B,1             ;INIT 'B' TO BIT POSITION 0
3084  4B6A  7B                4.0   TRANS1: MOV     A,E             ;GET CHAR TO BE TRANSLATED
3085  4B6B  A0                4.0           ANA     B               ;SEE IF BIT POSITION IN 'B' IS SET
3086  4B6C  CA  72  4B       10.0           JZ      TRANS2          ;DO NEXT BIT POSITION IF NOT SET
3087  4B6F  79                4.0           MOV     A,C             ;GET PREVIOUS RESULT OF 'OR'
3088  4B70  B6                7.0           ORA     M               ;'OR' IN NEW POSITION
3089  4B71  4F                4.0           MOV     C,A             ;SAVE RESULT
3090
3091  4B72  78                4.0   TRANS2: MOV     A,B
3092  4B73  07                4.0           RLC                     ;POSITION MASK TO NEXT BIT
3093  4B74  47                4.0           MOV     B,A
3094  4B75  D8               12.0           RC                      ;EXIT WHEN ALL POSITIONS DONE
3095  4B76  23                6.0           INX     H               ;POINT TO NEXT TABLE ENTRY
3096  4B77  C3  6A  4B       10.0           JMP     TRANS1          ;PROCESS NEXT BIT
3097        0000                    .END
```

```
A      =%0007          ABCNT   4A86          ABCNTR  4A87          ACRCEC  487A  G
ACRCEE  4880           ACRCFI  4A6D          ACRCHR  4AA0          ACRCOC  4868  G
ACRCOE  486E           ACRCP   4AA1          ACRCPE  486F          ACRCPO  485B
ACRCTB  4A65           ACRC1   4A3D          ACRC2   4A50          ACRC3   4A5C
ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002          ARAIDF= 0098
ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91          B      =%0000
BADST  = 0090          BIT0   = 0001          BIT1   = 0002          BIT15  = 8000
BIT2   = 0004          BIT3   = 0008          BIT4   = 0010          BIT5   = 0020
BIT6   = 0040          BIT7   = 0080          BIT8   = 0100          BIT9   = 0200
BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E30          BRKXCT= 4F00
BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24          BYTEL   4F23
C      =%0001          CACRC1  4881          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 0088          CBYTL  = 008A          CCTLWD  4A84          CDATBL  4AEF
CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C
CHPTIE= 0028           CHOTIE= 0020          CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027           CKACRC  47B1          CKAC4   4804          CKAC4C  481A  G
CKA1    47BD  G        CKA2    47CF  G       CKA3    47E1  G       CKA3A   47E7
CKA3B   47F2           CKA3C   47FE  G       CKA4    483D  G       CKA5    4820
CKCRC   488F           CKCRC1  489D          CKCRC2  4934          CKC1    48A9  G
CKC2    488B  G        CKECC   4732          CKECCC  4742  G       CKECC1  47AB
CKHACK  45CC           CKLOP  = 2815          CKPAD   4551          CKPADA  455D
CKPAD1  45C1           CKRESI  493C          CKRES1  49CB          CKR1    4948  G
CKR1A   4959           CKR2    496C  G       CKR5    4987  G       CKSCLK  4473
CKSC1   4482           CLACRC  4A16          CLCRC   4A02          CLECC   4B33
CLKCNT  4A8D           CLKCRC  4AA5          CLKCTL= 00F0          CLKON   4A8B
CLKSYS  4B27           CLOCK   4F26          CMC0H  = 0099          CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CRCCHR  4A98          CRCCNT  4A9A          CRCFIN  4A0D
CRCPE   4922           CRCPEC  492D  G       CRCPEE  4933          CRCPO   490E
CRCPOC  491B  G        CRCPOE  4921          CRCWRD= 0018          CRC1    49EE
CRC2    49F9           CRC2A   48CC          CRC3    49FE          CRC4    48DB
CRC5    48F0  G        CSAVE   4F9D          CSRLH  = 0091          CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082
C.     = 0001          C.AVAI= 0080          C.DP   = 0008          C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002          D      =%0002
DATACT= 00D0           DATPE   4664          DATPEC  466F  G       DATPEE  4675
DATPO   4650           DATPOC  465D  G       DATPOE  4663          DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB          DELOOP  4717          DELP1   4724
DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010          DPNUM   4A8A          DSAVE   4F9E
DSE    = 0006          DTEND   4B15          D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          EABCNT  4A88
ECCBAD= 0042           ECCCHR  4A97          ECCCK   4748          ECCCK1  475D  G
```

```
ECCCK2  476D            ECCCNT  4A8C            ECCCOR= 0019            ECCOK = 0041
ECCPE   4799            ECCPEC  47A4  G         ECCPEE  47AA            ECCPO   4785
ECCPOC  4792  G         ECCPOE  4798            ECCTBL  4B60
ECCTST= 000E            ECC1    4B4E            EDATA = 0095            EOTCLR= 0003
ERFLG   4F93            ERLP  = 2809            ERLPA = 280F            ERLPB = 2812
ERLPE = 280C            ERNUM   4F90            ERRCNT= 00D6            ESAVE   4F9F
E.ACRC= 0010            E.AMT = 0020            E.CDP = 0080            E.CRC = 0080
E.PNTR= 0008            E.RPE = 0040            E.STEC= 0001            E.TTEC= 0002
E.UNC = 0004            FIFORD= 006A            FORMAT  4F25            FORMT   4A89
FWDTST= 0061            GCRID = 0089            GCRSET= 0002            GENACR  4A1E
GENCRC  49D8            GENDAT  4AAA            GENECC  4B3D            GENOUT  4AC6
GOODTM= 0092            H     =%0004            HAVACR  4A9D            HAVCRC  4A9C
HAVRCH  4A9E            HLSAVE  4FA0            IE    = 0008            IMBYTC  4A85
IMPADC  4AA6            IMRESC  4AA4            INTSTA= 00E0            ITERA   4F9A
I.PWR = 0020            I.RMPE= 0040            I5.5  = 0010            I6.5  = 0020
I7.5  = 0040            KCALL = 005F            KCLR  = 007B            KDEP  = 003F
KENAB = 0078            KEXAM = 003E            KEYBRD= 00C8            KEY1  = 0078
KEY10 = 006D            KEY11 = 006E            KEY12 = 006F            KEY13 = 005C
KEY14 = 005D            KEY15 = 005E            KEY16 = 005F            KEY17 = 003C
KEY18 = C03D            KEY19 = 003E            KEY2  = 0079            KEY20 = 003F
KEY3  = 007A            KEY4  = 007B            KEY5  = 0074            KEY6  = 0075
KEY7  = 0076            KEY8  = 0077            KEY9  = 006C            KINTA = 006F
KLDAD = 003D            KN0   = 003C            KN1   = 005C            KN2   = 005D
KN3   = 005E            KN4   = 006C            KN5   = 006D            KN6   = 006E
KN7   = 0074            KN8   = 0075            KN9   = 0076            KU2   = 0079
KU3   = 007A            KU8   = 0077            L     =%0005            LASTCR  4AA9
LBLANK= 000F            LCE   = 000B            LCH   = 000C            LCL   = 000D
LCP   = 000E            LC0   = 0000            LC1   = 0001            LC2   = 0002
LC3   = 0003            LC4   = 0004            LC5   = 0005            LC6   = 0006
LC7   = 0007            LC8   = 0008            LC9   = 0009            LDLEDA= 00CA
LDLEDB= 00CB            LDLEDC= 00CC            LDLEDD= 00CD            LDLEDE= 00CE
LDLEDF= 00CF            LKDIAG= 2800            LKKBD = 004C            LKKEY = 0049
LKLWMG= 0058            LKLWMP= 0055            LKLWPG= 0052            LKLWPP= 004F
LKMOD7= 0046            LKOPR = 0046            LPFLG   4F94            LPNUM   4F92
M     =%0006            MBSEL = 00E0            MB.A  = 0008            MB.B  = 0004
MEMTOP= 4FFF            MINUS = 000A            MM    = 8000            MSE   = 0008
MSGN  = 0028            MTACLR= 0000            MT.ARA= 0020            MT.CPE= 0080
MT.DSE= 0001            MT.FWD= 0040            MT.INH= 0008            MT.LWR= 0004
MT.MOT= 0002            MT.NWT= 0080            MT.PEC= 0040            MT.PSB= 0004
MT.PS0= 0001            MT.PS1= 0002            MT.REV= 0020            MT.WRT= 0010
MT.Z  = 0008            M.ATA = 0080            M.CAPE= 0020            M.CONT= 0080
M.DEM = 0020            M.EBL = 0010            M.EXC = 0008            M.FAIL= 0008
M.ILR = 0010            M.INIT= 0010            M.OCC = 0020            M.ONLI= 0001
M.PE  = 0040            M.PORT= 0080            M.RDEN= 0002            M.RDPE= 0008
M.RUN = 0004            M.SCLK= 0001            M.TRA = 0040            M.UNIT= 0007
M.WCLK= 0040            M.WCLN= 0080            M.WREN= 0080            M5.5  = 0001
M6.5  = 0002            M7.5  = 0004            NOTCAP= 0088            OKAY  = 00FF
OPRRAM= 4300            OPSTRT= 0058            OPVER = 0040            PADCNT= 00D5
PADCN0  457D  G         PADCRC= 0080            PADPE   45AF            PADPEC  45BA  G
PADPEE  45C0            PADPO   459B            PADPOC  45A8  G         PADPOE  45AE
PARITY  4A76            PAR1    4A7F            PBIT    4AA2            PDCRC   4AA8
PDIAG = 0048            PEID  = 008A            PENAB = 004C            PESET = 0001
PL    = 00B1            PRDD  = 004C            PRENF = 009C            PS    = 00B2
PSTAT   0048            PSW   =%0009            P.AMTP= 0001            P.BCTC= 0040
```

P.CMDP= 0020 P.INTE= 0080 P.LCS = 0040 P.LWR = 0020
P.RDP = 0002 P.RPEN= 0004 P.RPST= 0002 P.RPOE= 0020
P.RP1E= 0010 P.RP2E= 0020 P.RP3E= 0010 P.SING= 0080
P.STAT= 0002 P.STPE= 0080 P.TACH= 0008 P.TUPR= 0010
P.WCSP= 0004 P.WDS = 0040 P.WFLP= 0001 P.WPEN= 001C
P.WPOE= 0008 P.WP1E= 0004 P.WP2E= 0008 P.WP3E= 0004
P.5VOK= 0002 QUE = 281B QUEM = 281E QUIT 46A8
QUIT1 46BD G QUIT2 46DD G QUIT3 46F2 G QUIT4 46E3
RAMT = 0010 RARA = 0006 RARAI = 0004 RCHBDO= 0048
RCHBD1= 0047 RCHOK = 0046 RCHRIM 4A9B RCHTST= 000C
RCLRT = 000D RCMD = 000B RCMLP = 0003 RCONT = 0080
RDATA = 0017 RDCLK = 0010 RDON = 0011 READG = 0007
REND = 0014 REQST = 2806 RESCHR= 00D1 RESCNT 4AA3
RESPE 49B9 RESPEC 49C4 G RESPEE 49CA RESPO 49A5
RESPOC 49B2 G RESPOE 49B8 REVTST= 0064 REWIND= 0004
RFIFOL= 0008 RGCLK = 0002 RGCRI = 0003 RIBG = 0001
RILL = 0012 RINST = 000C RMCTST= 0008 RMK2 = 0013
RNOP = 0000 RPADCT 4A9F RPATH = 0001 RPBAD = 0044
RPCHI = 0001 RPCLK = 0003 RPCTL = 0009 RPEI = 0002
RPFAIL= C000 RPF1 = 009D RPF2 = 009E RPOK = 0043
RPOSTN= 0016 RPSTA = 0015 RRCMT = 000A RSTAT = 0002
RTIEB = 000A RTIER = 0030 RTM = 0005 RUNKI = 0009
RUPTST= 005E RWDUNL= 0005 R.AMT = 0001 R.BOP = 0008
R.DATA= 0040 R.DON = 0002 R.DRDY= 0010 R.END = 0010
R.ILL = 0004 R.JVOK= 0004 R.MK2 = 0008 R.PLOD= 0008
R.PLO0= 0010 R.PLO1= 0020 R.POST= 0020 R.STNM= 0002
R.STOP= 0004 R.STPC= 0001 R.TBJN= 0080 R.TSTD= 0040
R.VOK = 0080 R00H = 0081 R00L = 0080 R01H = 0083
R01L = 0082 R02H = 0085 R02L = 0084 R03H = 0087
R03L = 0086 R04H = 0089 R04L = 0088 R05H = 008B
R05L = 008A R06H = 008D R06L = 008C R07H = 008F
R07L = 008E R10H = 0091 R10L = 0090 R11H = 0093
R11L = 0092 R12H = 0095 R12L = 0094 R13H = 0097
R13L = 0096 R14H = 0099 R14L = 0098 R15H = 009B
R15L = 009A R16H = 009D R16L = 009C R17H = 009F
R17L = 009E R7.5 = 0010 SCHREX 4889 SCKB1 4465
SCKGON 445E SCKWT 4436 SCKWTC 4453 G SCKWTE 445C
SCKWT1 4443 SCLKBI 442B SDDRA 4A8E SDDRAT 4A91
SDDRB 4A8F SDDRBT 4A92 SDDRC 4A90 SDDRCT 4A93
SELCLR= 0000 SETATA= 00A1 SETDDR 4ACA SHL2 4B22
SHL4 4B1E SHR2 4B19 SHR4 4B15 SID = 0080
SINTST 4A96 SOD = 0080 SOE = 0040 SP =%0008
SSCLK = 0040 SSTEP = 0005 STACK = 4FFF STATRM= 4F20
STPCT 4F20 STRSP = 5000 SWMCST 4A94 SXFER 4A95
TADR00= 0080 TADR01= 0081 TADR02= 0082 TADR03= 0083
TADR04= 0084 TADR05= 0085 TADR06= 0086 TADR07= 0087
TADR10= 0088 TADR11= 0089 TADR12= 008A TADR13= 008B
TAMT = 0044 TASEL = 0080 TCMD = 0040 TC.FWD= 0040
TC.INH= 0008 TC.LWR= 0004 TC.REV= 0020 TC.WRT= 0010
TEMP 4F99 TMF = 0099 TMRDY = 0040 TPBCNT 4AA7
TRANS 4B68 TRANS1 4B6A TRANS2 4B72 TRKENA= 00D2
TSET = 2 3 TSTEND= 2818 TSTS = 0040 TST1 4300
TST1A 43DF TST1B 43EA TST1BC 4410 G TST1C 4407
TST1D 4419 TST1DC 4422 G TST1L 4318 TST1NB 4311

```
TST1ST  4305            TST1XC  4632  G         TUSEL0= 00D1            TUSEL1= 00D2
TU78  = 0010            T.ATH0= 0001            T.ATH1= 0002            T.BOT = 0004
T.EOT = 0002            T.FPT = 0001            T.NTHR= 0004            T.ONL = 0020
T.PES = 0008            T.PSBJ= 0020            T.PSOJ= 0008            T.PS1J= 0010
T.RDY = 0080            T.RDY0= 0040            T.RWD = 0010            T.SCLK= 0002
T1RST   4378            T1RSTA  43A8            T1RSTB  43B1            T1RSTC  43C9
UIBG  = 00A1            VALFC   4F98            VALTB   4F95            VELTST= 005B
WDP.P = 0010            WMCCTL= 00D3            WMCERR= 00DA            WMCSTA= 00D0
WRTCLK= 0000            WRTDAT= 00D3            W.ACRC= 0004            W.CRC = 0008
W.DIAG= 0002            W.DONN= 0040            W.ECC = 0000            W.ENAB= 0080
W.ERR = 0020            W.FMT = 0070            W.GCR = 0010            W.LEFT= 0004
W.ONES= 0020            W.RESI= 0002            W.REV = 0004            W.ROME= 0010
W.RST = 0001            W.SKIP= 000F            W.WRIT= 0008            W.XFER= 0020
X     =%000A            XFB4C   4600  G         XFRBIT  4489            XFRB1   4490
XFRB2   44AE            XFRB2C  44C4  G         XFRB3   44CD            XFRB3A  44D4
XFRB3C  44EA  G         XFRB4   44F3            XFRB4C  4609            XFRB4D  4528
XFRB4P  4638            XFRB4R  453B            XFRB5   4676            XFRB5C  469F  G
XFRERR  45FB            XFRGON  44A6  G         XFRRC1  45F2  G         XFRRDY  44F9
X.DONN= 0080            X.ENAB- 0040            X.PEPE= 0002            X.ROME= 0001
X.WCLK= 0001            Y     -%000B            .     = 4B7A


 ERRORS DETECTED:  0

 *WMC4.A78/PTP,WMC4=NLIST,PARAM,MACRO,LIST,WMC4
 RUN-TIME: 5 8 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:07 SEQ 0188
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  WMC5 - WMC 'PE' MODE RECORD FORMATTING (XMC ENABLED)
                                .SBTTL  TEST 1 - WRITE MICRO ECC, ACRC, CRC TESTS
                                ;ID     WMC5-WRITE MICRO CONTROLLER PART #5
1332  4300                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1333                            ;*'PE' FORMAT BYTE ASSEMBLY TESTS
1334  4300                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1335                            ;*THIS TEST IS DESIGNED TO TEST THE WRITE MICROCONTROLLER ECC
1336                            ;*LOGIC.  A KNOWN DATA PATTERN IS FED INTO THE BYTE ASSEMBLY LOGIC FROM
1337                            ;*THE DIAGNOSTIC DATA REGISTERS - 'DDR' - TO PROVIDE AN 8 BIT WIDE
1338                            ;*REPLACEMENT FOR MASSBUS DATA IN IMAGE MODE.
1339                            ;*
1340                            ;*THE BYTE ASSEMBLY LOGIC WILL 'BREAK UP' THE INPUT DATA (8 BITS)
1341                            ;*INTO THE DESIRED BYTE FORMAT TO BE WRITTEN ON TAPE.
1342                            ;*
1343                            ;*THIS TEST WILL CHECK THE OUTPUT OF THE WRITE MICRO TO BE CORRECT
1344                            ;*IN REGARD TO THE ECC CHARACTERS GENERATED AND THE PROPER LOCATION
1345                            ;*OF THESE CHARACTERS IN THE DATA STREAM.
1346  4300                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1347                            ;*BGNTST
1348                            ;*  INIT THE TEST
1349                            ;*  SET FORMAT MODE TO 5 (IMAGE)
1350                            ;*  SET TRANSFER LENGTH TO 0 BYTES
1351                            ;*  BGNDO
1352                            ;*  :  UPDATE LENGTH OF TRANSFER BY 1 BYTE
1353                            ;*  :  SET DATA PATTERN INDEX TO START
1354                            ;*  :  RESTART WMC + SET CLOCK CONTROL TO SINGLE-STEP SYSTEM
1355                            ;*  :  INIT COUNTER CONTROL
1356                            ;*  :  CLEAR ECODE COUNTER
1357                            ;*  :  CLEAR PROGRAM FLAGS AND COUNTERS
1358                            ;*  :  SET THE DDRCTL WORD TO 'OUT'-ENABLES DATA INJECTION TO WMC
1359                            ;*  :  GENERATE A SET OF DATA INTO DDR
1360                            ;*  :  INIT SIMULATED 'ECC' BYTE AND 'CRC' BYTE
1361                            ;*  :  CLOCK SYSTEM THROUGH RESTART CODE (WMC ROM PROGRAM)
1362                            ;*  :  CALL SUBROUTINE 'LKMOD7' (OPR U-CODE) TO LOAD THE BYTE COUNTER
1363                            ;*  :  LOAD RESCHR WORD WITH DATA FROM MOD7 SUBROUTINE-MOD7 REMAINDER
1364                            ;*  :  CLEAR PAD COUNTER-NOT USED IN 'PE' DATA TRANSFERS
1365                            ;*  :  CALCULATE PROPER NUMBER OF 'PAD' BYTES TO EXPECT FOR CURRENT SIZE OF DATA XFR
1366                            ;*  :  LOAD DATACTL WORD FOR FORMAT MODE #5 (IMAGE)
1367                            ;*  :  START WMC ROM PGM BY SETTING WMCTL 'ENABLE' BITS AND 'WRITE' BIT
1368                            ;*  :  INIT TIMEOUT COUNTER
1369                            ;*  :  BGNDO
1370                            ;*  :  :  CLOCK THE SYSTEM
1371                            ;*  :  :  IF WMCSTA HAS NO 'XFER' BIT SET
1372                            ;*  :  :  :  THEN-UPDATE TIMEOUT COUNTER
1373                            ;*  :  :  :  :  IF TIMEOUT DETECTED
```

```
1374                    ;*  :  :  :  :  :  THEN-REPORT LACK OF ACTIVITY IN SPECIFIED TIME
1375                    ;*  :  :  :  :  :  ELSE-CONTINUE
1376                    ;*  :  :  :  :  :  ENDIF
1377                    ;*  :  :  :  :  ELSE-GET WMCSTA DATA AND SAVE FOR LATER TESTS
1378                    ;*  :  :  :  :  :  BGNDO
1379                    ;*  :  :  :  :  :  :  CLOCK SYSTEM TO FINISH 1ST NIBBLE-'XFER' GOES AWAY
1380                    ;*  :  :  :  :  :  :  IF > 20 CLOCKS
1381                    ;*  :  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' BIT ON 1ST NIBBLE
1382                    ;*  :  :  :  :  :  :  :  ELSE-CONTINUE
1383                    ;*  :  :  :  :  :  :  ENDIF
1384                    ;*  :  :  :  :  :  :  DO UNTIL 'XFER' CLEARS OR TIMEOUT
1385                    ;*  :  :  :  :  :  ENDDO
1386                    ;*  :  :  :  :  BGNDO
1387                    ;*  :  :  :  :  :  CLOCK SYSTEM TO BEGIN 2ND NIBBLE ASSEMBLY
1388                    ;*  :  :  :  :  :  IF > 128 CLOCKS
1389                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO SET 'XFER' FOR 2ND NIBBLE
1390                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1391                    ;*  :  :  :  :  :  ENDIF
1392                    ;*  :  :  :  :  :  DO UNTIL 'XFER' SETS OR TIMEOUT
1393                    ;*  :  :  :  :  ENDDO
1394                    ;*  :  :  :  :  UPDATE COUNT OF 'ZFER' STATUS RECEIVED
1395                    ;*  :  :  :  :  GET PARITY BIT ASSOCIATED WITH BYTE ASSEMBLED IN WRTDAT
1396                    ;*  :  :  :  :  BGNDO
1397                    ;*  :  :  :  :  :  CLOCK SYSTEM TO END 2ND NIBBLE
1398                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1399                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' TO FINISH 2ND NIBBLE
1400                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1401                    ;*  :  :  :  :  :  ENDIF
1402                    ;*  :  :  :  :  ENDDO
1403                    ;*  :  :  :  :  CLOCK SYSTEM TWICE TO 'SET' INTERNAL STATUS FOR ASSEMBLED BYTE
1404                    ;*  :  :  :  :  IF WMCSTA INDICATES A SPECIAL COUNTER
1405                    ;*  :  :  :  :  :  THEN-DETERMINE WHICH CHARACTER
1406                    ;*  :  :  :  :  :  :  IF CHARACTER IS 'ACRC'
1407                    ;*  :  :  :  :  :  :  :  THEN-REPORT 'ACRC' SENT IN PE MODE-ILLEGAL
1408                    ;*  :  :  :  :  :  :  :  ELSE-CONTINUE
1409                    ;*  :  :  :  :  :  :  ENDIF
1410                    ;*  :  :  :  :  :  :  IF CHARACTER IS 'RESID'
1411                    ;*  :  :  :  :  :  :  :  THEN-REPORT 'RESID' PRESENTED IN PE MODE-ILLEGAL
1412                    ;*  :  :  :  :  :  :  :  ELSE-CONTINUE
1413                    ;*  :  :  :  :  :  :  ENDIF
1414                    ;*  :  :  :  :  :  :  EXIT DO LOOP
1415                    ;*  :  :  :  :  :  ELSE-CONTINUE
1416                    ;*  :  :  :  :  ENDIF
1417                    ;*  :  :  :  :  UPDATE DATA BYTE RECEIVED COUNT FOR 'ECC' CHECKING
1418                    ;*  :  :  :  :  IF ALL DATA BYTES RECEIVED FROM WMC
1419                    ;*  :  :  :  :  :  THEN-CHECK 'PAD' BYTE RECEIVED
1420                    ;*  :  :  :  :  :  :  IF WRTDAT DATA=0 (VALID 'PAD' CHARACTER)
1421                    ;*  :  :  :  :  :  :  :  THEN-CONTINUE
1422                    ;*  :  :  :  :  :  :  :  ELSE-REPORT INVALID 'PAD' BYTE ASSEMBLED IN 'PE' MODE
1423                    ;*  :  :  :  :  :  :  ENDIF
1424                    ;*  :  :  :  :  :  :  IF PARITY BIT ON 'PAD' BYTE IS OK
1425                    ;*  :  :  :  :  :  :  :  THEN-CONTINUE
1426                    ;*  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON 'PAD' BYTE
1427                    ;*  :  :  :  :  :  :  ENDIF
```

```
                                ;*  :  :  :  :  :  ADD 'PAD' BYTE TO SIMULATED 'ECC' BYTE
                                ;*  :  :  :  :  :  IF ALL EXPECTED 'PAD' CHARACTERS RECEIVED
                                ;*  :  :  :  :  :  :  THEN-EXIT DO LOOP
                                ;*  :  :  :  :  :  :  ELSE-WAIT FOR TRANSLATOR TO TAKE THE 'SUBGROUP' FROM WMC
                                ;*  :  :  :  :  :  :  BGNDO
                                ;*  :  :  :  :  :  :  :  CLOCK THE SYSTEM
                                ;*  :  :  :  :  :  :  :  DO UNTIL TRANSLATOR IS CLOCKED ENOUGH TIMES TO RESPOND
                                ;*  :  :  :  :  :  :  ENDDO
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  ELSE-CHECK DATA BYTE ASSEMBLED IN WRTDAT
                                ;*  :  :  :  :  :  IF BYTE ASSEMBLED=EXPECTED BYTE
                                ;*  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  ELSE-REPORT PE BYTE ASSEMBLY FAULT
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  IF PARITY ON ASSEMBLED BYTE IS OK
                                ;*  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON DATA BYTE ASSEMBLED
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  ADD DATA BYTE TO SIMULATED 'ECC' WORD
                                ;*  :  :  :  :  :  ADD ALSO TO SIMULATED 'CRC' WORD
                                ;*  :  :  :  :  :  IF NECESSARY TO CHECK BYTE COUNTER
                                ;*  :  :  :  :  :  :  THEN-GET BYTECT DATA
                                ;*  :  :  :  :  :  :  :  IF BYTE COUNT=0 (AS EXPECTED)
                                ;*  :  :  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  :  :  ELSE-REPORT BYTE COUNTER NOT=0 WHEN EXPECTED
                                ;*  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  EXIT DO LOOP
                                ;*  :  :  :  :  ENDIF
                                ;*  :  :  :  ENDIF
                                ;*  :  :  ENDIF
                                ;*  :  :  DO UNTIL ALL 'XFER' BYTES RECEIVED FOR THIS DATA XFR
                                ;*  :  ENDDO
                                ;*  :  DO UNTIL ALL DATA PATTERNS/TRANSFERS HAVE BEEN TESTED
                                ;*  ENDDO
                                ;*ENDTST
4300                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC5 MICRO TEST 01
                                ;*WMC5 MICRO ERROR 01
                                ;*WMC5-WMC 'PE' FORMAT TEST
                                ;*M8959, M8958
                                ;*WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
                                ;*TEST ABORTED!
                                ;*
                                ;*WMC5 MICRO TEST 01
                                ;*WMC5 MICRO ERROR 02
                                ;*WMC5-WMC 'PE' FORMAT TEST
                                ;*M8959, M8958
                                ;*WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
                                ;*TEST ABORTED.
```

```
1479                                    ;*
1480                                    ;*WMC5 MICRO TEST 01
1481                                    ;*WMC5 MICRO ERROR 03
1482                                    ;*WMC5-WMC 'PE' FORMAT TEST
1483                                    ;*M8959, M8958
1484                                    ;*WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1485                                    ;*TEST ABORTED.
1486                                    ;*
1487                                    ;*WMC5 MICRO TEST 01
1488                                    ;*WMC5 MICRO ERROR 04
1489                                    ;*WMC5-WMC 'PE' FORMAT TEST
1490                                    ;*M8959, M8958
1491                                    ;*WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1492                                    ;*
1493                                    ;*WMC5 MICRO TEST 01
1494                                    ;*WMC5 MICRO ERROR 05
1495                                    ;*WMC5-WMC 'PE' FORMAT TEST
1496                                    ;*M8959, M8958
1497                                    ;*WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1498                                    ;*BYTE-SCLK COUNT = LLL
1499                                    ;*DATA FORMAT = MM
1500                                    ;*SKIP COUNT  = NN
1501                                    ;*TEST ABORTED!
1502                                    ;*
1503                                    ;*WMC5 MICRO TEST 01
1504                                    ;*WMC5 MICRO ERROR 06
1505                                    ;*WMC5-WMC 'PE' FORMAT TFST
1506                                    ;*M8959, M8958
1507                                    ;*WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1508                                    ;*BYTE-SCLK COUNT = LLL
1509                                    ;*DATA FORMAT = MM
1510                                    ;*SKIP COUNT  = NN
1511                                    ;*TEST ABORTED!
1512                                    ;*
1513                                    ;*WMC5 MICRO TEST 01
1514                                    ;*WMC5 MICRO ERROR 07
1515                                    ;*WMC5-WMC 'PE' FORMAT TEST
1516                                    ;*M8959, M8958
1517                                    ;*WMC PRESENTED AN 'ACRC' CHARACTER IN 'PE' MODE - ILLEGAL
1518                                    ;*BYTE-SCLK COUNT = LLL
1519                                    ;*DATA FORMAT = MM
1520                                    ;*SKIP COUNT  = NN
1521                                    ;*
1522                                    ;*WMC5 MICRO TEST 01
1523                                    ;*WMC5 MICRO ERROR 10
1524                                    ;*WMC5-WMC 'PE' FORMAT TEST
1525                                    ;*M8959, M8958
1526                                    ;*WMC PRESENTED 'RESID' BYTE IN 'PE' MODE - ILLEGAL
1527                                    ;*BYTE-SCLK COUNT = LLL
1528                                    ;*DATA FORMAT = MM
1529                                    ;*SKIP COUNT  = NN
1530                                    ;*
1531                                    ;*WMC5 MICRO TEST 01
1532                                    ;*WMC5 MICRO ERROR 11
```

```
1533                                    ;*WMC5-WMC 'PE' FORMAT TEST
1534                                    ;*M8959, M8958
1535                                    ;*PAD CHARACTER BYTE INCORRECT
1536                                    ;*ACTUAL = NNNN
1537                                    ;*EXPECTED = NNNN
1538                                    ;*
1539                                    ;*WMC5 MICRO TEST 01
1540                                    ;*WMC5 MICRO ERROR 12
1541                                    ;*WMC5-WMC 'PE' FORMAT TEST
1542                                    ;*M8959, M8958
1543                                    ;*PARITY BIT ON PAD SHOULD =0 AND WAS -1
1544                                    ;*ACTUAL = NNNN
1545                                    ;*
1546                                    ;*WMC5 MICRO TEST 01
1547                                    ;*WMC5 MICRO ERROR 13
1548                                    ;*WMC5-WMC 'PE' FORMAT TEST
1549                                    ;*M8959, M8958
1550                                    ;*PARITY BIT ON PAD WAS =0 AND SHOULD BE -1
1551                                    ;*ACTUAL = NNNN
1552                                    ;*
1553                                    ;*WMC5 MICRO TEST 01
1554                                    ;*WMC5 MICRO ERROR 14
1555                                    ;*WMC5-WMC 'PE' FORMAT TEST
1556                                    ;*M8959, M8958
1557                                    ;*XMC FAILED TO FINISH CLOCKING SO NEXT WMC CHAR CAN BE SENT
1558                                    ;*TEST ABORTED!
1559                                    ;*
1560                                    ;*WMC5 MICRO TEST 01
1561                                    ;*WMC5 MICRO ERROR 15
1562                                    ;*WMC5-WMC 'PE' FORMAT TEST
1563                                    ;*M8959, M8958
1564                                    ;*WMC SET 'XFER' ONLY - AFTER ALL DATA AND PAD CHARACTERS OUTPUT
1565                                    ;*
1566                                    ;*WMC5 MICRO TEST 01
1567                                    ;*WMC5 MICRO ERROR 16
1568                                    ;*WMC5-WMC 'PE' FORMAT TEST
1569                                    ;*M8959, M8958
1570                                    ;*WMC 'PE'' BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
1571                                    ;*BYTE-SCLK COUNT = LLL
1572                                    ;*DATA FORMAT = MM
1573                                    ;*SKIP COUNT  = NN
1574                                    ;*ACTUAL = NNNN
1575                                    ;*EXPECTED = NNNN
1576                                    ;*
1577                                    ;*WMC5 MICRO TEST 01
1578                                    ;*WMC5 MICRO ERROR 17
1579                                    ;*WMC5-WMC 'PE' FORMAT TEST
1580                                    ;*M8959, M8958
1581                                    ;*PARITY BIT ON DATA BYTE SHOULD =0 AND WAS =1
1582                                    ;*ACTUAL = NNNN
1583                                    ;*
1584                                    ;*WMC5 MICRO TEST 01
1585                                    ;*WMC5 MICRO ERROR 20
1586                                    ;*WMC5-WMC 'PE' FORMAT TEST
```

```
1587                                    ;*M8959, M8958
1588                                    ;*PARITY BIT ON DATA WAS -0 AND SHOULD BE =1
1589                                    ;*ACTUAL = NNNN
1590                                    ;*
1591                                    ;*WMC5 MICRO TEST 01
1592                                    ;*WMC5 MICRO ERROR 21
1593                                    ;*WMC5-WMC 'PE' FORMAT TEST
1594                                    ;*M8959, M8958
1595                                    ;*WMC BYTE COUNTER EXPECTED TO BE = 0 - ALL DATA BYTES RECEIVED FROM WMC
1596                                    ;*BYTE-SCLK COUNT = LLL
1597                                    ;*DATA FORMAT = MM
1598                                    ;*SKIP COUNT  = NN
1599                                    ;*ACTUAL = NNNN
1600                                    ;*
1601                                    ;*WMC5 MICRO TEST 01
1602                                    ;*WMC5 MICRO ERROR 22
1603                                    ;*WMC5-WMC 'PE' FORMAT TEST
1604                                    ;*M8959, M8958
1605                                    ;*SPECIAL 'PE' TERMINATING 'CRC' CHARACTER INCORRECT
1606                                    ;*ACTUAL = NNNN
1607                                    ;*EXPECTED = NNNN
1608                                    ;*
1609                                    ;*WMC5 MICRO TEST 01
1610                                    ;*WMC5 MICRO ERROR 23
1611                                    ;*WMC5-WMC 'PE' FORMAT TEST
1612                                    ;*M8959, M8958
1613                                    ;*WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
1614                                    ;*ACTUAL = NNNN
1615                                    ;*
1616                                    ;*WMC5 MICRO TEST 01
1617                                    ;*WMC5 MICRO ERROR 24
1618                                    ;*WMC5-WMC 'PE' FORMAT TEST
1619                                    ;*M8959, M8958
1620                                    ;*WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED - ACTUAL = 'XFER' COUNT
1621                                    ;*ACTUAL = NNNN
1622                                    ;*BYTE-SCLK COUNT - LLL
1623                                    ;*DATA FORMAT = MM
1624                                    ;*SKIP COUNT  = NN
1625                                    ;*
1626                                    ;*WMC5 MICRO TEST 01
1627                                    ;*WMC5 MICRO ERROR 25
1628                                    ;*WMC5-WMC 'PE' FORMAT TEST
1629                                    ;*M8959, M8958
1630                                    ;*WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED IN PE MODE
1631                                    ;*ACTUAL = NNNN
1632                                    ;*EXPECTED = NNNN
1633                                    ;*BYTE-SCLK COUNT - LLL
1634                                    ;*DATA FORMAT = MM
1635                                    ;*SKIP COUNT  = NN
1636                                    ;*
1637                                    ;*WMC5 MICRO TEST 01
1638                                    ;*WMC5 MICRO ERROR 26
1639                                    ;*WMC5-WMC 'PE' FORMAT TEST
1640                                    ;*M8959, M8958
```

```
                                        ;*PARITY BIT ON ECC SHOULD =0 AND WAS =1
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC5 MICRO TEST 01
                                        ;*WMC5 MICRO ERROR 27
                                        ;*WMC5-WMC 'PE' FORMAT TEST
                                        ;*M8959, M8958
                                        ;*PARITY BIT ON ECC WAS =0 AND SHOULD BE =1
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC5 MICRO TEST 01
                                        ;*WMC5 MICRO ERROR 28
                                        ;*WMC5-WMC 'PE' FORMAT TEST
                                        ;*M8959, M8958
                                        ;*WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
1658  4300                              $
 (1)                                    ; ***************************************************************************
1659
1660  4300                              TST1:   TESTX   @1
 (1)  4300  3E  01            7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1661                                    ;%WMC5-WMC 'PE' FORMAT TEST
1662                                    ;&M8959, M8958
1663
1664                                    ;INIT FORMAT CODE COUNTER TO 5 - FORMAT CODE = IMAGE MODE
1665
1666  4305  3E  05            7.0       TST1ST: MVI     A,@5            ;LOAD THE IMAGE MODE FORMAT CODE
1667  4307  32  4F  48        13.0               STA     FORMT           ;SAVE FORMAT CODE
1668  430A  AF                4.0               XRA     A
1669  430B  32  66  48        13.0               STA     TPBCNT          ;CLEAR THE TEST PASS BYTE COUNT
1670
1671                                    ;HERE TO UPDATE THE NUMBER OF BYTES TO USE THIS TEST PASS
1672
1673  430E  3A  66  48        13.0      TST1NBC: LDA     TPBCNT          ;GET THE BYTE COUNT
1674  4311  3C                4.0               INR     A
1675  4312  32  66  48        13.0               STA     TPBCNT          ;SAVE THE NEW BYTE COUNT TO USE
1676
1677                                    ;FOLLOWING IS A DUMMY HOST REQUEST FOR UNIT SELECT INFO...
1678                                    ;THIS IS NEEDED TO SIGNAL THE HOST THAT THE TEST IS STILL RUNNING.
1679                                    ;THIS WHOLE TEST TAKES ABOUT A MIN. OR SO TO RUN!
1680
1681  4315                                      REQ     7
 (1)  4315  CD  06  28       18.0                 CALL    REQST
 (1)  4318  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (1)  4319  00  00                                .WORD                   ;SYSTEM "" COUNT
 (1)  431B  00  00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  431D  00                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  431E  07                                    .BYTE   7               ;REQUEST CODE
1682
1683                                    ;INIT COUNTER CONTROL FOR BYTE COUNTER AND ECODE COUNTER
1684
1685                                    ;RESTART WRITE MICROCONTROLLER ROM PROGRAM
```

```
1686
1687  431F  3E  FF        7.0  TST1L:  MVI   A,@377          ;-1 TO OFFSET NUMBER
1688  4321  32  4F  48    13.0          STA   DPNUM           ;INIT DATA PATTERN OFFSET SO 1ST WILL BE '0'
1689  4324  3E  01        7.0          MVI   A,W.RST         ;GET 'RST' BIT
1690  4326  D3  D3       10.0          OUT   WMCCTL          ;RESTART WMC
1691  4328  3E  05        7.0          MVI   A,SSTEP         ;SET THE SYSTEM TO SINGLE STEP MODE
1692  432A  32  4A  48    13.0          STA   CCTLWD          ;SAVE THE CONTROL WORD
1693  432D  D3  F0       10.0          OUT   CLKCTL          ;WRITE THE CLOCK CONTROL WORD
1694  432F  AF            4.0          XRA   A               ;FINISH THE WMC RESTART
1695  4330  D3  D3       10.0          OUT   WMCCTL
1696  4332  3E  30        7.0          MVI   A,@60
1697  4334  D3  D7       10.0          OUT   CNTCTL          ;SELECT BYTE COUNTER
1698  4336  3E  70        7.0          MVI   A,@160
1699  4338  D3  D7       10.0          OUT   CNTCTL          ;SELECT THE PAD COUNTER
1700  433A  3E  B0        7.0          MVI   A,@260
1701  433C  D3  D7       10.0          OUT   CNTCTL          ;SELECT ECODE COUNTER
1702
1703                                ;CLEAR ERROR CODE COUNTER
1704  433E                          CLRECT                  ;CLEAR ECODE COUNTER
 (1)  433E  AF            4.0                XRA   A       ;CLEAR THE ACCUMULATOR
 (1)  433F  D3  D6       10.0                OUT   ERRCNT  ;CLEAR BITS 7-0
 (1)  4341  D3  D6       10.0                OUT   ERRCNT  ;CLEAR BITS 15-8
1705
1706  4343  32  51  48    13.0          STA   ECCCNT         ;CLEAR 'ECC' CHECK COUNTER
1707  4346  32  5D  48    13.0          STA   ECCEXP         ;CLEAR ECC EXPECTED FLAG
1708  4349  32  52  48    13.0          STA   CLKCNT         ;CLEAR THE 'SCLK' COUNTER
1709  434C  32  4C  48    13.0          STA   ABCNT          ;CLEAR THE ASSEM. BYTE COUNTER
1710  434F  32  62  48    13.0          STA   HAVCRC         ;CLEAR THE HAVE CRC FLAG
1711  4352  32  4D  48    13.0          STA   EABCNT         ;
1712  4355  32  65  48    13.0          STA   CLKCRC         ;CLEAR THE EXTRA CLOCK-TO-CRC FLAG
1713  4358  32  68  48    13.0          STA   LASTCRC        ;CLEAR THE END 'CRC' FLAG
1714  435B  32  63  48    13.0          STA   RPADCT         ;CLEAR COUNT OF RECEIVED PAD CHAR.
1715  435E  32  6B  48    13.0          STA   XFRREC         ;CLEAR COUNTER FOR 'XFER' BYTES RECEIVED
1716
1717                          ;SET DDR CONTROL TO 'OUT' FOR DATA INJECTION INTO BAL LOGIC
1718
1719  4361  3E  88        7.0          MVI   A,@210         ;SET 'OUT' BITS
1720  4363  D3  DB       10.0          OUT   DDRCTL         ;LOAD DDR CONTROL
1721
1722  4365  CD  6C  48    18.0          CALL  GENDAT         ;GENERATE THE 1ST SET OF DATA
1723  4368  11  B1  48    10.0          LXI   D,CDATBL       ;LOAD THE ADDRESS OF 'EXPECTED' DATA TABLE
1724  436B  CD  F5  48    18.0          CALL  CLECC          ;INIT THE ECC WORD - 'ECCCHR'
1725  436E  CD  2B  48    18.0          CALL  CLCRC          ;INIT CRC WORD
1726
1727  4371  06  B4        7.0          MVI   B,180          ;CLOCK THE SYSTEM THRU THE RESTART AREA
1728  4373  CD  E9  48    18.0  T1RST:  CALL  CLKSYS        ;CLOCK THE SYSTEM
1729  4376  05            4.0          DCR   B
1730  4377  C2  73  43    10.0          JNZ   T1RST          ;STAY HERE TILL DONE COUNTING
1731
1732  437A  3A  66  48    13.0          LDA   TPBCNT         ;GET THE BYTE COUNT TO USE
1733  437D  32  23  4F    13.0          STA   BYTEL          ;SAVE THE BYTE COUNT LOW VALUE
1734  4380  AF            4.0          XRA   A
1735  4381  32  24  4F    13.0          STA   BYTEH          ;CLEAR THE BYTE COUNT HIGH VALUE
1736  4384  CD  46  00    18.0          CALL  LKMOD7         ;CALL MOD7 ROUTINE FROM OPR U-CODE
```

```
1737  4387  78                4.0          MOV     A,B             ;GET MOD7 REMAINDER FROM LKMOD7
1738  4388  32  69  48       13.0          STA     MOD7R           ;SAVE THE # RESIDUAL DATA BYTES EXPECTED
1739  438B  0F                4.0          RRC                     ;POSITION THE MOD7 NUMBER
1740  438C  0F                4.0          RRC
1741  438D  0F                4.0          RRC
1742  438E  32  6A  48       13.0          STA     RCHRIM          ;SAVE IT
1743  4391  D3  D1           10.0          OUT     RESCHR          ;LOAD THE RESID CHAR WORD IN WMC
1744  4393  3A  66  48       13.0          LDA     TPBCNT          ;GET THE BC AGAIN
1745  4396  90                4.0          SUB     B               ;CALC THE VALUE LOADED INTO THE BYTE COUNTER
1746  4397  3C                4.0          INR     A
1747  4398  32  4B  48       13.0          STA     IMBYTC          ;SAVE FOR LATER TESTING
1748
1749  439B  AF                4.0  T1RSTA: XRA     A
1750  439C  D3  D5           10.0          OUT     PADCNT          ;LOAD PAD COUNT 7-0
1751
1752  439E  AF                4.0  T1RSTB: XRA     A
1753  439F  D3  D5           10.0          OUT     PADCNT          ;LOAD PAD CCOUNTER 15-8
1754  43A1  3A  66  48       13.0          LDA     TPBCNT          ;GET # BYTES IN XFR
1755  43A4  E6  07            7.0          ANI     7
1756  43A6  FE  07            7.0          CPI     7               ;EXPECT 'ECC'
1757  43A8  C2  AE  43       10.0          JNZ     T1RSTD          ;NOT EXPECTING AN 'ECC' BYTE
1758  43AB  32  5D  48       13.0          STA     ECCEXP          ;SET 'ECC' BYTE EXPECTED FLAG
1759
1760  43AE  3A  69  48       13.0  T1RSTD: LDA     MOD7R           ;GET # BYTES IN 'RESID' GROUP
1761  43B1  4F                4.0          MOV     C,A             ;COPY FOR DOUBLE ADD
1762  43B2  06  00            7.0          MVI     B,0             ;TO GEN AN OFFSET INTO A TABLE
1763  43B4  21  BF  43       10.0          LXI     H,PEXPTB        ;POINT TO PAD EXPECTED TABLE
1764  43B7  09                10.0          DAD     B               ;MAKE THE OFFSET
1765  43B8  7E                 7.0          MOV     A,M             ;GET THE # OF EXPECTED PADS FOR XFR
1766  43B9  32  5C  48       13.0          STA     PADEXP          ;SAVE NUMBER OF EXPECTED 'PAD' BYTES IN 'PE'
1767  43BC  C3  C7  43       10.0          JMP     T1RSTC          ;JUMP OVER THE TABLE
1768
1769  43BF  04                     PEXPTB: .BYTE   4               ;4 PADS IF NO 'RESID' GRP & EXPECT ECC
1770  43C0  03                             .BYTE   3               ; 1 BYTE IN 'RESID' GROUP= 3 PADS
1771  43C1  02                             .BYTE   2               ; 2 BYTES IN RES GRP = 2 PADS FOR PE
1772  43C2  01                             .BYTE   1               ; 3 BYTES = 1 PAD
1773  43C3  04                             .BYTE   4               ; 4 BYTES = 4 PADS
1774  43C4  03                             .BYTE   3
1775  43C5  02                             .BYTE   2
1776  43C6  04                             .BYTE   4               ;EXPECTING AN ECC CHAR SO WILL HAVE 4 PADS
1777
1778                                ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1779
1780  43C7  3E  50             7.0  T1RSTC: MVI     A,$50           ;LOAD IMAGE FORMAT CODE
1781  43C9                         ROUT    R02H                    ;ALSO SAVE IN CAS REG 02
 (1)  43C9  D3  85           10.0          OUT     R02H                    ;WRITE AC INTO R02H
 (1)  43CB  7F                4.0          MOV     A,A                     ;RETRY LINK
1782  43CC  D3  D0           10.0          OUT     DATACTL         ;LOAD DATA CONTROL WORD
1783  43CE  3E  04             7.0          MVI     A,4             ;SET RECORD COUNT TO 1
1784  43D0                         ROUT    R02L
 (1)  43D0  D3  84           10.0          OUT     R02L                    ;WRITE AC INTO R02L
 (1)  43D2  7F                4.0          MOV     A,A                     ;RETRY LINK
1785
1786                                ;ENABLE WRITE MICROCONTROLLER - START OF DATA XFR - WRITE
```

```
1787
1788  43D3  3E  C8        7.0  TST1A:  MVI   A,W.ENAB.X.ENAB!W.WRITE  ;SELECT 'WMC EN' + 'WRITE' (PE MODE)
1789  43D5  D3  D3       10.0          OUT   WMCCTL                    ;LOAD WRITE MICROCONTROLLER CONTROL
1790  43D7  AF            4.0          XRA   A
1791  43D8  32  50  48   13.0          STA   CLKON                     ;CLEAR 'SCLK' ON FLAG
1792
1793  43DB  06  FF        7.0          MVI   B,@377                    ;SET SINGLE STEP CLOCK 'TIMEOUT' LIMIT TO MAX POS. COUNT
1794  43DD  48            4.0          MOV   C,B                       ;COPY FOR THE 'XFER' TIMEOUT COUNTER
1795
1796  43DE  CD  74  44   18.0  TST1B:  CALL  CKSCLK                    ;ISSUE SINGLE STEP CLOCK
1797  43E1  DA  1F  43   10.0          JC    TST1L                     ;JUMP BACK IF ERROR DETECTED
1798  43E4  DB  D0       10.0          IN    WMCSTA                    ;GET WMC STATUS WORD
1799  43E6  E6  20        7.0          ANI   W.XFER                    ;SAVE ONLY 'XFER' BIT
1800  43E8  C2  8A  44   10.0          JNZ   XFRBIT                    ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
1801  43EB  3A  5A  48   13.0          LDA   SXFER                     ;GET THE SOFTWARE XFER FLAG
1802  43EE  A7            4.0          ANA   A
1803  43EF  CA  FB  43   10.0          JZ    TST1C                     ;JUMP IF XFER NOT SET
1804  43F2  AF            4.0          XRA   A                         ;CLEAR SOFTWARE FLAG
1805  43F3  32  5A  48   13.0          STA   SXFER                     ;FOR NEXT PASS
1806  43F6  06  7F        7.0          MVI   B,@177                    ;SETUP A WATCHDOG COUNT
1807  43F8  C3  AF  44   10.0          JMP   XFRB2                     ;PROCESS THE 'XFER' BIT DETECTED
1808
1809                                ;HERE TO SEE IF 'SCLK' TIMEOUT HAPPENED
1810
1811  43FB  05            4.0  TST1C:  DCR   B                       ;COUNT CLOCK CYCLE
1812  43FC  C2  14  44   10.0          JNZ   TST1D                     ;TIMEOUT? - JUMP IF OK
1813  43FF  3A  68  48   13.0          LDA   LASTCRC
1814  4402  A7            4.0          ANA   A
1815  4403  C2  ED  46   10.0          JNZ   QUIT                      ;EXIT IF END OF XFR
1816
1817  4406                              ERR   TST1L,TST1BC
 (1)                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4406  CD  09  28   18.0                CALL  ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0001                             MSGN  =     MSGN+1               ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4409  01                               .BYTE MSGN                ;MESSAGE NUMBER ID
 (1)  440A  00                               .BYTE
 (1)  440B  CD  15  28   18.0          TST1BC::      CALL  CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  440E  DA  1F  43   10.0                JC    TST1L               ;LOOP ADDRESS IF LOOP SPECIFIED
1818                            ;>WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1819                            ;<TEST ABORTED!
1820  4411  C3  18  28   10.0          JMP   TSTEND                    ;TERMINATE TESTING NOW! - FATAL ERROR
1821
1822                                ;HERE TO SEE IF 'XFER' BIT TIMEOUT HAPPENED
1823
1824  4414  0D            4.0  TST1D:  DCR   C                       ;DECREMENT 'XFER' WATCHDOG COUNTER
1825  4415  C2  DE  43   10.0          JNZ   TST1B                     ;BACK TO LOOP IF OK
```

```
1827  4418                                          ERR     TST1L,TST1DC
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4418  CD   09   28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  441B  02                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  441C  00                                           .BYTE
 (1)  441D  CD   15   28        18.0               TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4420  DA   1F   43        10.0                       JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1828                                    ;>WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1829                                    ;<TEST ABORTED!
1830  4423  C3   18   28        10.0               JMP     TSTEND
```

```
1833  4426                                  $
 (1)                                        ;*****************************************************************************
1834                                        ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
1835                                        ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
1836  4426                                  $
 (1)                                        ;*****************************************************************************
1837
1838  4426  3A  50  48        13.0  SCLKBIT: LDA     CLKON            ;SEE IF 'SCLK' WAS ON
1839  4429  A7                 4.0           ANA     A                ;SET CONDITION BITS
1840  442A  C0                12.0           RNZ                      ;EXIT IF CLOCK WAS SET-HAVE DATA ALREADY
1841  442B  CD  8C  48        18.0           CALL    SETDDR           ;THEN LOAD THE DATA INTO THE DDR REGS
1842  442E  C5                12.0           PUSH    B                ;SAVE REG B & C
1843  442F  06  14             7.0           MVI     B,20             ;GET A WATCHDOG COUNT
1844  4431  0E  FF             7.0  SCKWT:    MVI     C,@377
1845  4433  CD  E9  48        18.0  SCKWT1:   CALL    CLKSYS           ;CLOCK THE SYSTEM - WAIT FOR 'SCLK' TO GO AWAY
1846  4436  DB  D0             10.0           IN      WMCSTA           ;GET THE WMC STATUS WORD
1847  4438  E6  20             7.0           ANI     W.XFER           ;DID 'XFER' BIT SET?
1848  443A  CA  40  44         10.0           JZ      SCKWT2           ;NO - JUMP OVER
1849  443D  32  5A  48        13.0           STA     SXFER            ;YUP - SET THE FLAG
1850  4440  DB  C0             10.0  SCKWT2:   IN      DBUSSTA          ;GET THE DATA BUS STATUS INFO
1851  4442  E6  02             7.0           ANI     T.SCLK           ;SEE IF SCLK IS GONE YET
1852  4444  CA  5F  44         10.0           JZ      SCKGON           ;YUP - PROCEED
1853  4447  0D                 4.0           DCR     C                ;NO - SEE IF DONE WITH LOW COUNTER
1854  4448  C2  33  44         10.0           JNZ     SCKWT1           ;BACK TO LOOP
1855  444B  05                 4.0           DCR     B                ;YES! - DOWNCOUNT WATCHDOG
1856  444C  C2  31  44         10.0           JNZ     SCKWT            ;STAY IN LOOP TILL NO 'SCLK' OR DIE
1857  444F                                    ERR     SCKWTE,SCKWTC
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  444F  CD  09  28        18.0                   CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0003                            MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4452  03                              .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4453  00                              .BYTE
 (1)  4454  CD  15  28        18.0  SCKWTC::         CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4457  DA  5D  44        10.0                   JC      SCKWTE           ;LOOP ADDRESS IF LOOP SPECIFIED
1858                                    ;>WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1859                                    ;<TEST ABORTED!
1860  445A  C3  18  28        10.0           JMP     TSTEND           ;EXIT IF NOT LOOPING ON ERROR
1861
1862  445D  C1                10.0  SCKWTE:  POP     B
1863  445E  C9                10.0           RET
1864  445F  C1                10.0  SCKGON:  POP     B                ;ALL OK - NOW RESET REG B & C
1865  4460  C2  66  44         10.0           JNZ     SCKB1            ;JMP IF NOT TIME TO RELOAD DDR
1866
1867  4463  CD  6C  48        18.0           CALL    GENDAT           ;GENERATE NEW DDR DATA FOR INJECTION
1868
1869  4466  06  7F             7.0  SCKB1:   MVI     B,@177           ;GET NEW TIMEOUT COUNT
1870  4468  78                 4.0           MOV     A,B              ;COPY TO REG A
1871  4469  32  50  48        13.0           STA     CLKON            ;SET SCLK 'ON" FLAG
1872  446C  3A  52  48        13.0           LDA     CLKCNT           ;GET THE CURRENT SCLK COUNT
1873  446F  3C                 4.0           INR     A                ;UPDATE IT
1874  4470  32  52  48        13.0           STA     CLKCNT           ;SAVE THE NEW COUNT
1875  4473  C9                10.0           RET
```

```
                                        S
                                        ; ***************************************************************************
                                        ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
                                        ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
                                        ;       THE DDR.
                                        S
                                        ; ***************************************************************************

4474  CD  E9  48      18.0  CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
4477  DB  C0           10.0          IN      DBUSSTA         ;GET STATUS OF DATA BUS
4479  E6  02            7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
447B  C2  83  44       10.0          JNZ     CKSC1           ;JUMP IF SCLK IS ON
447E  AF                4.0          XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
447F  32  50  48       13.0          STA     CLKON
4482  C9               10.0          RET                     ;EXIT

4483  CD  26  44      18.0  CKSC1:  CALL    SCLKBIT         ;LOAD NEW DATA INTO DDR REGS
4486  DA  1F  43      10.0          JC      TST1L
4489  C9              10.0          RET                     ;EXIT

                                        S
                                        ; ***************************************************************************
                                        ;XFRBIT -- HERE WHEN WMC STATUS INDICATES 'XFER' BIT IS SET. ISSUE 20
                                        ;       MORE CLOCKS THEN CHECK WRITE DATA WORD.
                                        S
                                        ; ***************************************************************************

448A  DB  D0           10.0  XFRBIT: IN      WMCSTA          ;GET THE CURRENT WMC STATUS
448C  32  59  48       13.0          STA     SWMCSTA         ;SAVE IT FOR THE ECC CHECK (LATER)
448F  06  14            7.0          MVI     B,20            ;ALLOW UP TO 20. WMC CLOCKS TO DROP XFER
4491  CD  74  44       18.0  XFRB1:  CALL    CKSCLK          ;NEED TWO MORE CLOCK CYCLES BEFORE
4494  DA  1F  43       10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
4497  DB  D0           10.0          IN      WMCSTA          ;GET WMC STATUS
4499  E6  20            7.0          ANI     W.XFER          ;SAVE THE XFER BIT
449B  CA  A7  44       10.0          JZ      XFRGON          ;OK TO CONTINUE IF XFER IS GONE
449E  05                4.0          DCR     B               ;DECREMENT WATCHDOG COUNTER
449F  C2  91  44       10.0          JNZ     XFRB1           ;LOOP TIL =0 OR XFER IS GONE
44A2                                  ERR     TST1L,XFRGON
                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 44A2  CD  09  28      18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)       0004                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 44A5  04                                .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 44A6  00                                .BYTE
(1) 44A7  CD  15  28      18.0  XFRGON::      CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1) 44AA  DA  1F  43      10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                      ;>WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY


                                      ;HERE TO LOOK FOR 'XFER' TO SET FOR THE 2ND NIBBLE OF THIS BYTE

44AD  06  7F            7.0          MVI     B,@177
44AF  CD  74  44       18.0  XFRB2:  CALL    CKSCLK          ;SEE IF SCLK IS SET (ALSO CLOCK)
44B2  DA  1F  43       10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
44B5  DB  D0           10.0          IN      WMCSTA          ;GET WMC STATUS
```

```
1920  44B7  E6  20          7.0          ANI   W.XFER          ;SAVE ONLY XFER BIT
1921  44B9  C2  CE  44      10.0          JNZ   XFRB3           ;JUMP IF XFER IS SET - OK
1922  44BC  05               4.0          DCR   B               ;LOOK FOR TIMEOUT
1923  44BD  C2  AF  44      10.0          JNZ   XFRB2           ;GO BACK TO LOOP IF OK
1924  44C0                                ERR   TST1L,XFRB2C,3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44C0  CD  09  28      18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)                                            MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)       0005
 (1)  44C3  05                                  .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44C4  03                                  .BYTE 3
 (1)  44C5  CD  15  28      18.0          XFRB2C::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44C8  DA  1F  43      10.0                JC    TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1925                                ;>WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1926                                ;<TEST ABORTED!
1927  44CB  C3  18  28      10.0          JMP   TSTEND
1928
1929                                ;HERE TO LOOK FOR 'XFER' BIT TO GO AWAY TO INDICATE FULL BYTE IS AVAIL.
1930                                ;FROM THE 'WRTDAT' WORD
1931
1932  44CE  3A  6B  48      13.0  XFRB3:  LDA   XFRREC          ;GET CURRENT COUNT OF 'XFER' BYTES
1933  44D1  3C               4.0          INR   A               ;COUNT THIS ONE
1934  44D2  32  6B  48      13.0          STA   XFRREC          ;SAVE UPDATED COUNT
1935  44D5  06  14           7.0          MVI   B,20
1936  44D7  DB  E0          10.0          IN    INTSTA          ;GET THE DATA PARITY BIT
1937  44D9  32  5B  48      13.0          STA   SINTSTA         ;SAVE FOR PARITY CHECK
1938  44DC  CD  74  44      18.0  XFRB3A: CALL  CKSCLK          ;CLOCK THE SYSTEM TILL XFER BIT GOES AWAY
1939  44DF  DA  1F  43      10.0          JC    TST1L           ;JUMP IF ERROR DETECTED
1940  44E2  DB  D0          10.0          IN    WMCSTA          ;GET THE WMC STATUS
1941  44E4  E6  20           7.0          ANI   W.XFER          ;SAVE ONLY THE XFER BIT
1942  44E6  CA  FB  44      10.0          JZ    XFRB4           ;JUMP IF GONE - OK
1943  44E9  05               4.0          DCR   B               ;DOWNCOUNT WATCHDOG
1944  44EA  C2  DC  44      10.0          JNZ   XFRB3A          ;BACK TO LOOP IF OK
1945  44ED                                ERR   TST1L,XFRB3C,3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44ED  CD  09  28      18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)                                            MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)       0006
 (1)  44F0  06                                  .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44F1  03                                  .BYTE 3
 (1)  44F2  CD  15  28      18.0          XFRB3C::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44F5  DA  1F  43      10.0                JC    TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1946                                ;>WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1947                                ;<TEST ABORTED!
1948  44F8  C3  18  28      10.0          JMP   TSTEND
1949                                ;HERE TO TEST THE ASSEMBLED BYTE DATA AGAINST THE EXPECTED BYTE DATA
1950
1951  44FB  CD  E9  48      18.0  XFRB4:  CALL  CLKSYS          ;ALLOW THE BYTE COUNTER TO TICK
1952  44FE  CD  E9  48      18.0          CALL  CLKSYS          ;TO LATCH IT.
```

```
                                        $
                                        ;*****************************************************************************
                                        ;XFRRDY -- ROUTINE TO DETERMINE WHAT CHARACTER IS ON WMC OUTPUT WHEN
                                        ;       'XFER' IS SET.  ALSO, CHECKS FOR ERROR; BASED ON BYTE COUNTER.
                                        ;
                                        $
                                        ;*****************************************************************************

4501 3A 59 48      13.0 XFRRDY: LDA     SWMCSTA          ;GET WMC STATUS WORD
4504 E6 1E          7.0          ANI     @36              ;SAVE JUST THE SPECIAL CHAR INFO.
4506 CA 45 45      10.0          JZ      XFRB4D           ;NOTHING SPECIAL - CHECK DATA BYTE

4509 3A 59 48      13.0          LDA     SWMCSTA          ;GET THE STATUS AGAIN
450C E6 10          7.0          ANI     W.ECC            ;SEE IF ITS 'ECC' CHARACTER
450E C2 57 47      10.0          JNZ     CKECC            ;JUMP IF ITS 'ECC' - CHECK PE ECC CHAR

4511 3A 59 48      13.0          LDA     SWMCSTA          ;GET WMC STATUS WORD AGAIN
4514 E6 04          7.0          ANI     W.ACRC           ;SEE IF 'ACRC' CHARACTER READY
4516 CA 27 45      10.0          JZ      XFRDY2
4519                             ERR     TST1L,XFRDY1,3
                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
4519 CD 09 28      18.0                     CALL    ERLP             ;PROCESS ERROR - DO 2.3
     0007                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
451C 07                          .BYTE   MSGN             ;MESSAGE NUMBER ID
451D 03                          .BYTE   3
451E CD 15 28      18.0          XFRDY1::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
4521 DA 1F 43      10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
                                 ;>WMC PRESENTED AN 'ACRC' CHARACTER IN 'PE' MODE - ILLEGAL
4524 C3 2F 46      10.0          JMP     SCHREX

4527 3A 59 48      13.0 XFRDY2: LDA     SWMCSTA          ;GET STATUS AGAIN
452A E6 08          7.0          ANI     W.CRC            ;SEE IF ITS 'CRC' READY BIT
452C C2 CC 47      10.0          JNZ     CKCRC            ;GO TEST 'CRC' DATA BYTE

452F 3A 59 48      13.0          LDA     SWMCSTA          ;GET STATUS AGAIN
4532 E6 02          7.0          ANI     W.RESID          ;SEE IF 'RESID' READY BIT (IF NOT ITS FATAL!)
4534 CA 45 45      10.0          JZ      XFRB4D
4537                             ERR     TST1L,XFRDY3,3
                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
4537 CD 09 28      18.0                     CALL    ERLP             ;PROCESS ERROR - DO 2.3
     0008                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
453A 08                          .BYTE   MSGN             ;MESSAGE NUMBER ID
453B 03                          .BYTE   3
453C CD 15 28      18.0          XFRDY3::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
453F DA 1F 43      10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
                                 ;>WMC PRESENTED 'RESID' BYTE IN 'PE' MODE - ILLEGAL
4542 C3 2F 46      10.0          JMP     SCHREX



                                 ;HERE TO CHECK THE DATA BYTE JUST ASSEMBLED ( OULD BE A 'PAD' BYTE)

4545 3A 51 48      13.0 XFRB4D: LDA     ECCCNT           ;GET 'XFER' COUNT
4548 3C             4.0          INR     A                ;ADD 1
4549 32 51 48      13.0          STA     ECCCNT           ;SAVE THE UPDATED COUNT
454C E5            12.0          PUSH    H                ;SAVE H&L
```

```
1992  454D  21  66  48  10.0          LXI   H,TPBCNT        ;POINT TO THE AMT OF BYTES EXPECTED
1993  4550  3A  4C  48  13.0          LDA   ABCNT           ;THEN GET # BYTES ASSEMBLED BY THE WMC
1994  4553  BE              7.0      CMP   M               ;SEE IF DONE
1995  4554  E1             10.0      POP   H               ;RESET REG H & L
1996  4555  DA  41  46     10.0      JC    XFRB4C          ;CONTINUE TO COMPARE DATA BYTES
1997
1998
1999                        ;HERE TO CHECK THE PAD CHARACTERS EXPECTED
2000
2001  4558  3A  5C  48  13.0  CKPAD:  LDA   PADEXP          ;GET THE PAD COUNT EXPECTED
2002  455B  47               4.0      MOV   B,A             ;COPY THE COUNT
2003
2004  455C  3A  63  48  13.0  CKPADA: LDA   RPADCT          ;GET THE RECEIVED PAD COUNT #
2005  455F  B8               4.0      CMP   B               ;SAME AS EXPECTED (TIME FOR PHANTOM XFR)
2006  4560  CA  0E  46     10.0      JZ    CKHACK          ;GO CHECK FOR THE PHANTOM 'XFR'
2007  4563  3A  63  48  13.0          LDA   RPADCT          ;GET THE PAD CHARACTERS COUNT
2008  4566  3C               4.0      INR   A               ;UPDATE FOR THIS CHARACTER
2009  4567  32  63  48  13.0          STA   RPADCT          ;SAVE IT
2010  456A  AF               4.0      XRA   A               ;CLEAR THE ACCUMULATOR
2011  456B                            ROUT  EDATA           ;STORE THE EXPECTED PAD CHARACTER
 (1)  456B  D3  95          10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  456D  7F               4.0                MOV   A,A            ;RETRY LINK
2012  456E  DB  D3          10.0          IN    WRTDAT          ;GET THE ACTUAL PAD CHARACTER
2013  4570                            ROUT  ADATA           ;STORE THE ACTUAL PAD CHARACTER
 (1)  4570  D3  94          10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  4572  7F               4.0                MOV   A,A            ;RETRY LINK
2014  4573  A7               4.0          ANA   A               ;SET THE CONDITION BITS
2015  4574  CA  7C  45     10.0          JZ    PADCNO          ;CONTINUE IF ZERO
2016  4577                            ERRB  TST1L,PADCNO
 (1)                        ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4577  CD  12  28     18.0                CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0009                             MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  457A   09                             .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  457B   00                             .BYTE                 ;PRINT ROUTINE NUMBER
 (1)  457C  CD  15  28     18.0  PADCNO::       CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  457F  DA  1F  43     10.0                JC    TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2017                        ;>PAD CHARACTER BYTE INCORRECT
2018
```

```
2020
2021                                        ;HERE TO CHECK THE PARITY BIT ON THE PAD ASSEMBLED
2022
2023  4582  DB  D3        10.0        IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2024  4584                              ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  4584  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4586  7F             4.0                MOV     A,A             ;RETRY LINK
2025  4587  A7             4.0        ANA     A               ;SET THE CONDITION BITS
2026  4588  E4  9A  45     18.0        CPO     PADPO           ;CHECK FOR ODD PARITY
2027  458B  DA  1F  43     10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2028  458E  DB  D3         10.0        IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2029  4590  A7             4.0        ANA     A               ;SET THE CONDITION BITS
2030  4591  EC  AE  45     18.0        CPE     PADPE           ;CHECK FOR PARITY EVEN - BIT -1
2031  4594  DA  1F  43     10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2032  4597  C3  C0  45     10.0        JMP     CKPAD1          ;ON TO THE NEXT SECTION....
2033
2034                                        ;HERE TO CHECK FOR PARITY ODD - BIT =0
2035
2036  459A  3A  5B  48      13.0  PADPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2037  459D  E6  10           7.0        ANI     WDR.P           ;SAVE THE PARITY BIT
2038  459F  CA  AD  45      10.0        JZ      PADPOE          ;EXIT IF OK
```

```
2040  45A2                                          ERRA    PADPOE,PADPOC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  45A2  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000A                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45A5  0A                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45A6  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45A7  CD  15  28        18.0                  PADPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45AA  DA  AD  45        10.0                          JC      PADPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2041                                        ;>PARITY BIT ON PAD SHOULD =0 AND WAS =1
2042
2043  45AD  C9                   10.0  PADPCE: RET
2044
2045                                        ;HERE TO CHECK FOR PARITY BIT -1 (EVEN)
2046
2047  45AE  3A  5B  48           13.0  PADPE:  LDA     SINTSTA         ;GET THE BIT
2048  45B1  E6  10                7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2049  45B3  C0                   12.0          RNZ
2050  45B4                                      ERRA    PADPEE,PADPEC
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  45B4  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000B                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45B7  0B                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45B8  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45B9  CD  15  28        18.0                  PADPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45BC  DA  BF  45        10.0                          JC      PADPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2051                                        ;>PARITY EIT ON PAD WAS =0 AND SHOULD BE =1
2052
2053  45BF  C9                   10.0  PADPEE: RET
2054
2055  45C0  DB  D3               10.0  CKPAD1: IN      WRTDAT          ;GET THE PAD CHARACTER AGAIN
2056  45C2  CD  FF  48           18.0          CALL    GENECC          ;GO INCLUDE THE PAD CHARACTER IN THE ECC
2057
2058  45C5  3A  5C  48           13.0          LDA     PADEXP          ;GET THE EXPECTED COUNT OF PAD CHAR
2059  45C8  47                    4.0          MOV     B,A             ;SAVE
2060  45C9  3A  63  48           13.0          LDA     RPADCT          ;GET ACTUAL RECEIVED PAD COUNT
2061  45CC  B8                    4.0          CMP     B               ;SEE IF ALL RECEIVED
2062  45CD  C2  2F  46           10.0          JNZ     SCHREX          ;EXIT IF NOT DONE
2063
2064  45D0  06  19                7.0  XMCCK:  MVI     B,25            ;SET COUNTER FOR TRANSLATOR TO FINISH
2065
2066  45D2  0E  FF                7.0  XMCK1A: MVI     C,@377
2067
2068  45D4  CD  74  44           18.0  XMCK1:  CALL    CKSCLK          ;ISSUE SINGLE STEP CLOCK
2069  45D7  DA  1F  43           10.0          JC      TST1L           ;JUMP BACK IF ERROR DETECTED
2070  45DA  DB  D0               10.0          IN      WMCSTA          ;GET WMC STATUS WORD
2071  45DC  E6  20                7.0          ANI     W.XFER          ;SAVE ONLY 'XFER' BIT
2072  45DE  C2  8A  44           10.0          JNZ     XFRBIT          ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
2073  45E1  3A  5A  48           13.0          LDA     SXFER           ;GET THE SOFTWARE XFER FLAG
2074  45E4  A7                    4.0          ANA     A
2075  45E5  CA  F1  45           10.0          JZ      XMCK2           ;JUMP IF XFER NOT SET
2076  45E8  AF                    4.0          XRA     A               ;CLEAR SOFTWARE FLAG
2077  45E9  32  5A  48           13.0          STA     SXFER           ;FOR NEXT PASS
2078  45EC  06  7F                7.0          MVI     B,@177          ;SETUP A WATCHDOG COUNT
2079  45EE  C3  AF  44           10.0          JMP     XFRB2           ;PROCESS THE 'XFER' BIT DETECTED
```

```
2080
2081  45F1  0D                    4.0  XMCK2:  DCR   C               ;DOWNCOUNT FOR XMC
2082  45F2  C2  D4  45           10.0          JNZ   XMCK1           ;CLOCK AGAIN
2083
2084  45F5  05                    4.0  XMCK3:  DCR   B               ;DOWNCOUNT GROSS COUNT
2085  45F6  C2  D2  45           10.0          JNZ   XMCK1A          ;RESET OTHER COUNTER AND CONTINUE
2086
2087  45F9  3A  68  48           13.0          LDA   LASTCRC         ;TIME TO EXIT?
2088  45FC  A7                    4.0          ANA   A
2089  45FD  C2  ED  46           10.0          JNZ   QUIT            ;JUMP IF ALL DONE
2090  4600                                      ERR   TST1L,XMCK3C
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4600  CD  09  28           18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                              MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4603  0C                                .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4604  00                                .BYTE
 (1)  4605  CD  15  28           18.0          XMCK3C::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4608  DA  1F  43           10.0                JC    TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2091                          ;>XMC FAILED TO FINISH CLOCKING SO NEXT WMC CHAR CAN BE SENT
2092                          ;<TEST ABORTED!
2093  460B  C3  18  28           10.0          JMP   TSTEND
2094
2095
2096                          ;HERE IF A 'XFER' DATA BYTE WAS DETECTED AND ALL DATA BYTES WERE
2097                          ;RECEIVED AND ALL PAD BYTES REVEIVED...I.E. WE ARE ABOUT TO GIVE
2098                          ;AN EXTRA CLOCK TO THE CRC GENERATOR (CRC GROUP HAS 9 CHAR).
2099
2100  460E  3A  65  48           13.0  CKHACK: LDA   CLKCRC          ;GET THE EXTRA CLOCK-TO-CRC FLAG
2101  4611  A7                    4.0          ANA   A               ;SEE IF HERE BEFORE
2102  4612  C2  24  46           10.0          JNZ   XFRERR          ;YES - THEN NO MORE PHANTOM 'XFR'S' ALLOWED
2103  4615  3E  FF                7.0          MVI   A,@377          ;NO - THEN ALLOW ONLY THE ONE
2104  4617  32  65  48           13.0          STA   CLKCRC          ;UPDATE THE RECEIVED FLAG
2105  461A  3A  51  48           13.0          LDA   ECCCNT          ;GET THE ECC CHAR. COUNTER
2106  461D  3D                    4.0          DCR   A               ;ADJUST FOR ADDITION OF THE PHANTOM 'XFR'
2107  461E  32  51  48           13.0          STA   ECCCNT          ;RESTORE ADJUSTED COUNT
2108  4621  C3  2F  46           10.0          JMP   SCHREX          ;CONTINUE TO CLOCK THE WMC
2109
2110  4624                                    XFRERR: ERR   TST1L,XFB4C
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4624  CD  09  28           18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000D                              MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4627  0D                                .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4628  00                                .BYTE
 (1)  4629  CD  15  28           18.0          XFB4C:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  462C  DA  1F  43           10.0                JC    TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2111                          ;>WMC SET 'XFER' ONLY - AFTER ALL DATA AND PAD CHARACTERS OUTPUT
2112
2113  462F  06  7F                7.0  SCHREX: MVI   B,@177          ;RESET WATCHDOG TIMERS
2114  4631  48                    4.0          MOV   C,B
2115  4632  3A  6B  48           13.0          LDA   XFRREC          ;GET COUNT OF RECEIVED 'XFER' CHAR
2116  4635  FE  04                7.0          CPI   4               ;TIME TO EXTEND THE COUNT
2117  4637  DA  FB  43           10.0          JC    TST1C           ;NO - BACK TO NORMAL TIMEOUT LOOP
2118  463A  AF                    4.0          XRA   A               ;YUP
2119  463B  32  6B  48           13.0          STA   XFRREC          ;CLEAR THE COUNTER
```

```
2120  463E  C3  D0  45      10.0          JMP     XMCCK           ;ENTER SPECIAL 'HOLDING' LOOP
2121
2122                                ;GET ASSEMBLED BAL DATA BYTE FROM WRITE DATA WORDS
2123
2124  4641  3A  4C  48      13.0  XFRB4C: LDA     ABCNT           ;GET THE BYTE # UNDER TEST
2125  4644  3C                4.0          INR     A               ;ADD 1
2126  4645  32  4C  48      13.0          STA     ABCNT           ;SAVE THE BYTE NUMBER - UPDATED
2127  4648                               ROUT    R05L
 (1)  4648  D3  8A             10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  464A  7F                 4.0                   MOV     A,A             ;RETRY LINK
2128  464B  AF                 4.0          XRA     A
2129  464C                               ROUT    R05H
 (1)  464C  D3  8B             10.0                  OUT     R05H            ;WRITE AC INTO R05H
 (1)  464E  7F                 4.0                   MOV     A,A             ;RETRY LINK
2130  464F  EB                 4.0          XCHG                    ;PUT 'EXPECTED' POINTER IN H & L
2131  4650  7E                 7.0          MOV     A,M             ;GET CORRECT BAL BYTE
2132  4651                               ROUT    EDATA           ;SAVE 'EXPECTED' DATA
 (1)  4651  D3  95             10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4653  7F                 4.0                   MOV     A,A             ;RETRY LINK
2133  4654  DB  D3             10.0          IN      WRTDAT          ;GET ASSEMBLED DATA BYTE
2134  4656                               ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 (1)  4656  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4658  7F                 4.0                   MOV     A,A             ;RETRY LINK
2135  4659  BE                 7.0          CMP     M               ;EXPECTED = ACTUAL?
2136  465A  EB                 4.0          XCHG                    ;RESET H & L REGS
2137  465B  13                 6.0          INX     D               ;POINT TO NEXT 'EXPECTED' ENTRY
2138  465C  CA  6A  46         10.0          JZ      TST1XC          ;JUMP OVER ERROR IF DATA BYTE OK
2139  465F  3A  4C  48         13.0          LDA     ABCNT           ;GET THE COUNT OF ASSEM. BYTES AT    LURE
2140  4662  32  4D  48         13.0          STA     EABCNT          ;SAVE FOR THE LOOP CONTROL
2141  4665                               ERRB    DELOOP,TST1XC,3
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4665  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000E                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4668  0E                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4669  03                                   .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  466A  CD  15  28     18.0          TST1XC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  466D  DA  3C  47     10.0                  JC      DELOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2142                          ;>WMC 'PE' BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
2143
2144                          ;HERE TO CHECK THE PARITY BIT ON THE DATA ASSEMBLED
2145
2146  4670  DB  D3             10.0  XFRB4P: IN      WRTDAT          ;GET BYTE ON WMC OUTPUT
2147  4672                               ROUT    ADATA           ;SAVE IT
 (1)  4672  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4674  7F                 4.0                   MOV     A,A             ;RETRY LINK
2148  4675  A7                 4.0          ANA     A
2149  4676  E4  88  46         18.0          CPO     DATPO           ;CHECK FOR ODD PARITY ON DATA
2150  4679  DA  1F  43         10.0          JC      TST1L           ;ERROR RETURN
2151
2152  467C  DB  D3             10.0          IN      WRTDAT          ;GET DATA AGAIN
2153  467E  A7                 4.0          ANA     A
2154  467F  EC  9C  46         18.0          CPE     DATPE           ;CHECK FOR EVEN PARITY DATA
2155  4682  DA  1F  43         10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2156  4685  C3  AE  46         10.0          JMP     XFRB5           ;ON TO THE NEXT SECTION....
```

```
2157
2158                                    ;HERE IS THE ODD PARITY DATA CHECK
2159
2160  4688  3A 5B 48      13.0  DATPO:  LDA     SINTSTA         ;GET COPY OF INTSTA WORD
2161  468B  E6 10            7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2162  468D  CA 9B 46        10.0          JZ      DATPOE          ;EXIT IF ODD PARITY ON DATA OK
2163  4690                                ERRA    DATPOE,DATPOC
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4690  CD 0F 28      18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000F                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4693  0F                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4694  00                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4695  CD 15 28      18.0          DATPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4698  DA 9B 46        10.0                  JC      DATPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2164                                    ;>PARITY BIT ON DATA BYTE SHOULD =0 AND WAS =1
2165
2166  469B  C9                10.0  DATPOE: RET
2167
2168                                    ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2169
2170  469C  3A 5B 48          13.0  DATPE:  LDA     SINTSTA         ;GET THE BIT
2171  469F  E6 10              7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2172  46A1  C0                12.0          RNZ
2173  46A2                                ERRA    DATPEE,DATPEC
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  46A2  CD 0F 28      18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0010                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A5  10                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46A6  00                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46A7  CD 15 28      18.0          DATPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46AA  DA AD 46        10.0                  JC      DATPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2174                                    ;>PARITY BIT ON DATA WAS =0 AND SHOULD BE =1
2175
2176  46AD  C9                10.0  DATPEE: RET
2177
2178  46AE  DB D3              10.0  XFRB5:  IN      WRTDAT          ;GET THE DATA AGAIN
2179  46B0  CD FF 48          18.0          CALL    GENECC          ;CALCULATE THE ECC
2180  46B3  CD FE 47          18.0          CALL    GENCRC          ;ADD DATA TO CRC CHAR
2181
2182  46B6  3A 4B 48          13.0          LDA     IMBYTC          ;GET LOADED BYTE COUNT
2183  46B9  3D                 4.0          DCR     A               ;TEST TO SEE IF > 1
2184  46BA  CA 2F 46          10.0          JZ      SCHREX          ;DON'T TEST BYTE COUNT =0 IF LOADED WITH 1
2185
2186  46BD  E5                12.0          PUSH    H
2187  46BE  21 4C 48          10.0          LXI     H,ABCNT         ;POINT TO # ASSEMB. BYTES
2188  46C1  3A 69 48          13.0          LDA     MOD7R           ;GET MOD7 REMAINDER
2189  46C4  47                 4.0          MOV     B,A             ;TEMP STORE IT
2190  46C5  3A C5 48          13.0          LDA     TPBCNT          ;GET # EXPECTED BYTES
2191  46C8  90                 4.0          SUB     B               ;CALC WHEN BYTE COUNTER SHOULD RUN OUT
2192  46C9  BE                 7.0          CMP     M               ;SEE IF BYTE COUNTER SHOULD =0
2193  46CA  E1                10.0          POP     H               ;RESET REG H & L
2194  46CB  C2 2F 46          10.0          JNZ     SCHREX          ;BACK TO LOOP IF MORE BYTES TO DO
2195
2196  46CE  DB D4              10.0          IN      BYTCNT          ;DUMMY READ OF BYTE COUNTER
```

```
 2197  46D0  DB  D4          10.0          IN      BYTCNT
 2198
 2199                                ;TIME TO CHECK THE BYTE COUNTER...
 2200
 2201  46D2  DB  D4          10.0          IN      BYTCNT          ;GET BYTE COUNTER 7-0
 2202  46D4  47               4.0          MOV     B,A             ;TEMP STORE
 2203  46D5  DB  D4          10.0          IN      BYTCNT          ;GET BYTE CNT 15-8
 2204  46D7  78               4.0          MOV     A,B
 2205  46D8  A7               4.0          ANA     A
 2206  46D9  CA  E4  46       10.0          JZ      XFRB5C          ;JUMP IF BYTE COUNT OK
 2207  46DC                                ROUT    ADATA           ;STORE THE ACTUAL BYTE COUNTER 7-0
  (1)  46DC  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  46DE  7F               4.0                  MOV     A,A             ;RETRY LINK
 2208  46DF                                ERRA    TST1L,XFRB5C,3
  (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
  (1)  46DF  CD  0F  28       18.0                         CALL    ERLPA           ;PROCESS ERROR - DO 2.3
  (1)      0011                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46E2  11                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46E3  03                                            .BYTE   3               ;PRINT ROUTINE NUMBER
  (1)  46E4  CD  15  28       18.0                 XFRB5C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46E7  DA  1F  43       10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
 2209                                ;>WMC BYTE COUNTER EXPECTED TO BE = 0 - ALL DATA BYTES RECEIVED FROM WMC
 2210
 2211
 2212  46EA  C3  2F  46       10.0          JMP     SCHREX          ;BACK TO CLOCK LOOP FOR OTHER CHARACTERS
 2213
 2214  46ED                               S
  (1)                                   ; ***************************************************************************
 2215                                   ;HERE IF ALL DATA, ECC, CRC
 2216  46ED                               S
  (1)                                   ; ***************************************************************************
 2217
 2218  46ED  DB  D3          10.0  QUIT:   IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
 2219  46EF                                ROUT    ADATA
  (1)  46EF  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  46F1  7F               4.0                  MOV     A,A             ;RETRY LINK
 2220  46F2  21  5F  48       10.0          LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
 2221  46F5  BE               7.0          CMP     M               ;ACTUAL=SIMULATED?
 2222  46F6  CA  02  47       10.0          JZ      QUIT1           ;JUMP IF OK
 2223  46F9  7E               7.0          MOV     A,M             ;GET SIMULATED CRC CHAR.
 2224  46FA                                ROUT    EDATA
  (1)  46FA  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  46FC  7F               4.0                  MOV     A,A             ;RETRY LINK
 2225  46FD                                ERRB    TST1L,QUIT1
  (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
  (1)  46FD  CD  12  28       18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)      0012                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4700  12                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4701  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4702  CD  15  28       18.0                 QUIT1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4705  DA  1F  43       10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
 2226                                ;>SPECIAL 'PE' TERMINATING 'CRC' CHARACTER INCORRECT
 2227
 2228
```

```
2229
2230  4708  DB  D0          10.0  QUIT4:  IN      WMCSTA          ;GET THE WMC STATUS
2231  470A                        ROUT    ADATA           ;SAVE THE ACTUAL STATUS FOR PRINTOUT
 (1)  470A  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  470C  7F               4.0                MOV     A,A             ;RETRY LINK
2232  470D  E6  40           7.0          ANI     W.DONN          ;DONE?
2233  470F  CA  17  47       10.0         JZ      QUIT3           ;YES - EXIT THE TEST
2234  4712                         ERRA    TST1L,QUIT3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4712  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0013                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4715  13                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4716  00                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4717  CD  15  28       18.0         QUIT3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  471A  DA  1F  43       10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2235                              ;>WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
2236
2237
2238  471D  3A  66  48       13.0          LDA     TPBCNT          ;GET THE CURRENT BYTE COUNT USED
2239  4720  FE  26           7.0          CPI     DTEND-DATBL     ;DONE WITH ALL COMBINATIONS?
2240  4722  C2  0E  43       10.0         JNZ     TST1NBC         ;JUMP IF NOT AT END OF ROPE...
2241
2242  4725                         ENDTST  TST1ST
 (1)                               ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4725                         REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4725  CD  06  28       18.0                CALL    REQST
 (2)  4728  00                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  4729  00  00                   .WORD                   ;SYSTEM '*' COUNT
 (2)  472B  00  00                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  472D  00                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  472E  07                       .BYTE   7               ;REQUEST CODE
 (1)  472F  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4732  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4733  32  9A  4F       13.0         STA     ITERA           ;SAVE COUNT
 (1)  4736  F2  05  43       10.0         JP      TST1ST          ;DO TEST UNTIL TILL = 0
2243  4739  C3  18  28       10.0         JMP     TSTEND          ;ALL DONE
2244
2245                              ;HERE IS THE DATA COMPARE ERROR LOOP CONTROL ROUTINE
2246
2247                              ;WILL ENABLE THE TEST TO LOOP AT THE SPECIFIED BYTE IN ERROR
2248
2249  473C  3A  94  4F       13.0  DELOOP: LDA     LPFLG           ;SEE IF LOOP FLAG STILL SET
2250  473F  A7                    4.0          ANA     A
2251  4740  C2  49  47       10.0         JNZ     DELP1           ;JUMP IF LOOPING ON ERROR
2252  4743  32  4D  48       13.0         STA     EABCNT          ;NOT LOOPING - CLEAR THE FLAG
2253  4746  C3  AE  46       10.0         JMP     XFRB5           ;AND EXIT
2254
2255  4749  3A  4C  48       13.0  DELP1:  LDA     ABCNT           ;GET THE CURRENT COUNT OF ASSEMBLED BYTES
2256  474C  47                    4.0          MOV     B,A             ;TEMP STORE
2257  474D  3A  4D  48       13.0         LDA     EABCNT          ;GET THE BYTE COUNT AT THE DETECTED ERROR
2258  4750  B8                    4.0          CMP     B               ;SEE IF AT THE SAME BYTE COUNT
2259  4751  CA  1F  43       10.0         JZ      TST1L           ;THEN JUMP TO RESTART TEST IF SAME
2260  4754  C3  AE  46       10.0         JMP     XFRB5           ;ELSE, CONTINUE WITH THE TEST
```

```
                                        .SBTTL  SUBROUTINE CKECC
                                        $
                                        ; ***************************************************************************
                                        ;CKECC -- ROUTINE TO CHECK THE 'ECC' CHARACTER PRESENTED BY THE WMC
                                        ;       LOGIC AGAINST A CACULATED ECC BYTE. WILL REPORT AN ERROR IF
                                        ;       THE 'ECC' FLAG (IN WMCSTA WORD) COMES UP BEFORE OR AFTER THE
                                        ;       8TH ASSEMBLED BYTE FROM THE WMC OR IF AN ACTUAL ECC BYTE
                                        ;       ASSEMBLY ERROR WAS DETECTED.
                                        $
                                        ; ***************************************************************************
4757  3A  51  48      13.0  CKECC:  LDA     ECCCNT          ;GET 'XFER' CYCLE COUNT
475A                         ROUT    ADATA
475A  D3  94             10.0          OUT     ADATA           ;WRITE AC INTO ADATA
475C  7F                  4.0          MOV     A,A             ;RETRY LINK
475D  FE  07              7.0  CPI     7               ;READY FOR ECC CHAR?
475F  CA  6D  47         10.0  JZ      ECCCK           ;JUMP IF EXPECTED NOW
4762                         ERRA    TST1L,CKECCC,3
                             ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
4762  CD  0F  28       18.0          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
      0014                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
4765  14                     .BYTE   MSGN            ;MESSAGE NUMBER ID
4766  03                     .BYTE   3               ;PRINT ROUTINE NUMBER
4767  CD  15  28       18.0  CKECCC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
476A  DA  1F  43         10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                             ;>WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED - ACTUAL = 'XFER' COUNT


                             ;HERE TO CHECK THE ACTUAL FCC BYTE FROM THE WMC LOGIC

476D  21  5E  48         10.0  ECCCK:  LXI     H,ECCCHR        ;POINT TO 'IMAGE' OF HARDWARE ECC WORD
4770  7E                  7.0          MOV     A,M
4771                         ROUT    EDATA
4771  D3  95             10.0          OUT     EDATA           ;WRITE AC INTO EDATA
4773  7F                  4.0          MOV     A,A             ;RETRY LINK
4774  DB  D3             10.0  IN      WRTDAT          ;GET ACTUAL ECC WORD FROM WMC
4776                         ROUT    ADATA
4776  D3  94             10.0          OUT     ADATA           ;WRITE AC INTO ADATA
4778  7F                  4.0          MOV     A,A             ;RETRY LINK
4779  BE                  7.0  CMP     M               ;IS IT OK?
477A  CA  82  47         10.0  JZ      ECCCK1          ;JUMP IF SAME-AS EXPECTED
477D                         ERRB    TST1L,ECCCK1,3
                             ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
477D  CD  12  28         18.0          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
      0015                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
4780  15                     .BYTE   MSGN            ;MESSAGE NUMBER ID
4781  03                     .BYTE   3               ;PRINT ROUTINE NUMBER
4782  CD  15  28       18.0  ECCCK1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
4785  DA  1F  43       10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                             ;>WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED IN PE MODE

                             ;HERE TO CHECK THE PARITY BIT ON THE ECC ASSEMBLED

4788  DB  D3             10.0  ECCCK2: IN      WRTDAT          ;GET THE ECC BYTE AGAIN
```

```
2294  478A                                          ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  478A  D3  94          10.0                             OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  478C  7F               4.0                             MOV     A,A             ;RETRY LINK
2295  478D  A7               4.0                     ANA     A               ;SET THE CONDITION BITS
2296  478E  E4  A0  47       18.0                    CPO     ECCPO           ;CHECK FOR ODD PARITY
2297  4791  DA  1F  43       10.0                    JC      TST1L           ;JUMP IF ERROR DETECTED
2298  4794  DB  D3           10.0                    IN      WRTDAT          ;GET THE ECC BYTE AGAIN
2299  4796  A7               4.0                     ANA     A               ;SET THE CONDITION BITS
2300  4797  EC  B4  47       18.0                    CPE     ECCPE           ;CHECK FOR PARITY EVEN - BIT =1
2301  479A  DA  1F  43       10.0                    JC      TST1L           ;JUMP IF ERROR DETECTED
2302  479D  C3  C6  47       10.0                    JMP     CKECC1          ;ON TO THE NEXT SECTION....
2303
2304                                        ;HERE TO CHECK FOR PARITY ODD - BIT =0
2305
2306  47A0  3A  5B  48       13.0  ECCPO:   LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2307  47A3  E6  10            7.0           ANI     WDR.P           ;SAVE THE PARITY BIT
2308  47A5  CA  B3  47       10.0           JZ      ECCPOE          ;EXIT IF OK
2309  47A8                                  ERRA    ECCPOE,ECCPOC
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  47A8  CD  0F  28       18.0                   CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0016                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47AB  16                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47AC  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47AD  CD  15  28       18.0           ECCPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47B0  DA  B3  47       10.0                   JC      ECCPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2310                                        ;>PARITY BIT ON ECC SHOULD =0 AND WAS =1
2311
2312  47B3  C9                 10.0  ECCPOE: RET
2313
2314                                        ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2315
2316  47B4  3A  5B  48       13.0  ECCPE:   LDA     SINTSTA         ;GET THE BIT
2317  47B7  E6  10            7.0           ANI     WDR.P           ;SAVE THE PARITY BIT
2318  47B9  C0                12.0           RNZ
2319  47BA                                  ERRA    ECCPEE,ECCPEC
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  47BA  CD  0F  28       18.0                   CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0017                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47BD  17                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47BE  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47BF  CD  15  28       18.0           ECCPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47C2  DA  C5  47       10.0                   JC      ECCPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2320                                        ;>PARITY BIT ON ECC WAS =0 AND SHOULD BE =1
2321
2322  47C5  C9                 10.0  ECCPEE: RET
2323  47C6  CD  F5  48         18.0  CKECC1: CALL    CLECC
2324  47C9  C3  2F  46         10.0          JMP     SCHREX          ;GO BACK TO CLOCK LOOP
```

```
                                        .SBTTL  SUBROUTINE CKCRC
                                        $
                                        ; ***************************************************************************
                                        ;CKCRC -- ROUTINE TO CHECK THE WMC 'CRC' DATA BYTE.  CHECKS THAT THE
                                        ;         ACTUAL WMC 'CRC' CHARACTER IS THE SAME AS THE CALCULATED 'CRC'
                                        ;         CHARACTER.
                                        $
                                        ; ***************************************************************************
                                        
47CC 06 02          7.0 CKCRC:  MVI     B,2             ;SET CRC CHAR LIMIT TO 1
47CE 3A 62 48      13.0 CRC2A:  LDA     HAVCRC          ;GET CRC CHAR RECEIVED COUNT
47D1 3C             4.0         INR     A               ;ADD 1
47D2 32 62 48      13.0         STA     HAVCRC          ;SAVE UPDATED RECEIVED COUNT
47D5 FE 01          7.0         CPI     1               ;1ST CRC CHARACTER?
47D7 C2 E0 47      10.0         JNZ     CRC4
47DA 32 68 48      13.0         STA     LASTCRC         ;SET THE LAST CRC CHAR FLAG
47DD CD 33 48      18.0         CALL    CRCFIN          ;FINISH CRC CHAR SIMULATION-ONLY ON 1ST CHAR.
47E0 DB D3         10.0 CRC4:   IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
47E2                            ROUT    ADATA
(1) 47E2 D3 94     10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
(1) 47E4 7F         4.0                 MOV     A,A             ;RETRY LINK
47E5 21 5F 48      10.0         LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
47E8 BE             7.0         CMP     M               ;ACTUAL=SIMULATED?
47E9 CA F5 47      10.0         JZ      CRC5            ;JUMP IF OK
47EC 7E             7.0         MOV     A,M             ;GET SIMULATED CRC CHAR.
47ED                            ROUT    EDATA
(1) 47ED D3 95     10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
(1) 47EF 7F         4.0                 MOV     A,A             ;RETRY LINK
47F0                        ERRB    TST1L,CRC5
(1)                         ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
(1) 47F0 CD 12 28  18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
(1)      0018                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 47F3 18                             .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 47F4 00                             .BYTE                   ;PRINT ROUTINE NUMBER
(1) 47F5 CD 15 28  18.0             CRC5::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1) 47F8 DA 1F 43  10.0                     JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                            ;>WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
47FB C3 2F 46      10.0         JMP     SCHREX          ;ON TO THE NEXT SECTION....

                                        $
                                        ; ***************************************************************************
                                        ;CRC -- ROUTINE TO SIMULATE THE CRC GENERATOR IN THE TM78.  CALLED
                                        ;       WITH ACCUMULATOR CONTAINING DATA BYTE TO BE ADDED TO CRCWORD
                                        $
                                        ; ***************************************************************************

47FE CD 3C 48      18.0 GENCRC: CALL    PARITY          ;CALC. PARITY ON DATA IN ACCUMULATOR
4801 F5            12.0         PUSH    PSW
4802 C5            12.0         PUSH    B
4803 E5            12.0         PUSH    H
4804 21 5F 48      10.0         LXI     H,CRCCHR        ;POINT TO CRC CHAR STORAGE
4807 AE             7.0         XRA     M               ;XOR DATA IN ACCUMULATOR WITH PREVIOUS CRC 7 TO 0
4808 23             6.0         INX     H               ;POINT TO CRC PARITY BIT STORAGE
4809 47             4.0         MOV     B,A             ;SAVE RESULT OF XOR
```

```
2365  480A  3A 64 48   13.0          LDA   PBIT          ;GET PARITY OF DATA BYTE ON CALL
2366  480D  AE           7.0          XRA   M             ;XOR WITH CRC PARITY BIT
2367  480E  1F           4.0          RAR                 ;PUT PARITY BIT IN 'C' BIT POSITION
2368  480F  78           4.0          MOV   A,B           ;GET XOR OF 7 TO 0
2369  4810  32 61 48     13.0          STA   CRCOLD        ;SAVE FOR PE 'CRC' CHAR TEST
2370  4813  1F           4.0          RAR                 ;SHIFT RIGHT
2371  4814  D2 22 48     10.0          JNC   CRC2          ;COMPLIMENT BITS 5,4,3,+2 IF 'C' IS SET
2372
2373  4817  EE 3C        7.0  CRC1:   XRI   @74           ;COMPLIMENT BITS
2374  4819  32 5F 48     13.0          STA   CRCCHR        ;SAVE THE BYTE
2375  481C  3E 01        7.0          MVI   A,1           ;AND SET THE 'P' BIT TO A 1 FOR NEXT TIME
2376  481E  77           7.0          MOV   M,A           ;SAVE THE 1
2377  481F  C3 27 48     10.0          JMP   CRC3          ;EXIT
2378
2379  4822  32 5F 48     13.0  CRC2:   STA   CRCCHR        ;SAVE CRC CHARACTER
2380  4825  AF           4.0          XRA   A             ;CLEAR THE 'P' BIT
2381  4826  77           7.0          MOV   M,A           ;FOR THE NEXT TIME
2382  4827  E1          10.0  CRC3:   POP   H
2383  4828  C1          10.0          POP   B
2384  4829  F1          10.0          POP   PSW
2385  482A  C9          10.0          RET
2386
2387                                ;INITIALIZE CRC CHARACTER STORAGE
2388
2389  482B  AF           4.0  CLCRC:  XRA   A
2390  482C  32 5F 48     13.0          STA   CRCCHR
2391  482F  32 60 48     13.0          STA   CRCCHR+1
2392  4832  C9          10.0          RET
2393
2394                                ;TRANSFER COMPLETE... FINISH THE CRC GENERATION
2395
2396  4833  3A 61 48     13.0  CRCFIN: LDA   CRCOLD         ;GET CRC WORD
2397  4836  EE D7        7.0          XRI   @327          ;COMPLIMENT BITS 7,6,4,2,1,+0
2398  4838  32 5F 48     13.0          STA   CRCCHR
2399  483B  C9          10.0          RET
```

```
2401                                        .SBTTL  PARITY CALCULATION ROUTINE
2402                                        ;THIS SUBROUTINE CALCULATES ODD PARITY ON THE CONTENTS OF THE
2403                                        ;ACCUMULATOR AND SETS OR CLEARS THE MEMORY LOCATION 'PBIT AS REQUIRED.
2404
2405  483C  F5              12.0  PARITY: PUSH    PSW             ;SAVE ACCUMULATOR
2406  483D  A7               4.0          ANA     A               ;SET CONDITION BITS
2407  483E  3E 00            7.0          MVI     A,$00           ;CLEAR ACCUMULATOR
2408  4840  E2 45 48        10.0          JPO     PAR1            ;SKIP IF PARITY ODD
2409  4843  3E 01            7.0          MVI     A,$01           ;ELSE-LOAD ACC. WITH 1
2410  4845  32 64 48        13.0  PAR1:   STA     PBIT            ;SET UP THE P BIT
2411  4848  F1              10.0          POP     PSW             ;RESTORE ACCUMULATOR
2412  4849  C9              10.0          RET                     ;RETURN
2413                                      ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE
2414
2415  484A  00                          CCTLWD: .BYTE   0         ;IMAGE OF CLOCK CONTROL WORD
2416  484B  00                          IMBYTC: .BYTE   0         ;IMAGE OF BYTE COUNTER 7-0 BITS
2417
2418  484C  00                          ABCNT:  .BYTE   0         ;ASSEMBLED BYTE COUNTER
2419  484D  00                          EABCNT: .BYTE   0         ;BYTE COUNT AT A DETECTED ERROR
2420
2421  484E  00                          FORMT:  .BYTE   0         ;FORMAT CODE SELECTED FOR TEST
2422  484F  00                          DPNUM:  .BYTE   0         ;DATA PATTERN OFFSET NUMBER (0 TO 17)
2423  4850  00                          CLKON:  .BYTE   0         ;'SCLK' ON FLAG-USED TO TELL WHEN TO FILL DDR REGS
2424  4851  00                          ECCCNT: .BYTE   0         ;COUNT BY 7 FOR ECC CHARACTER CHECK
2425  4852  00                          CLKCNT: .BYTE   0         ;COUNT OF SCLKS GENERATED
2426
2427  4853  00                          SDDRA:  .BYTE   0         ;DDRA DATA ACTUALLY LOADED
2428  4854  00                          SDDRB:  .BYTE   0         ;DDRB DATA ACTUALLY LOADED
2429  4855  00                          SDDRC:  .BYTE   0         ;DDRC DATA ACTUALLY LOADED
2430
2431  4856  00                          SDDRAT: .BYTE   0         ;DDRA DATA FROM PGM TABLE
2432  4857  00                          SDDRBT: .BYTE   0         ;DDRB DATA FROM PGM TABLE
2433  4858  00                          SDDRCT: .BYTE   0         ;DDRC DATA FROM PGM TABLE
2434
2435  4859  00                          SWMCSTA: .BYTE  0         ;COPY OF THE WMC STATUS AT 'XFER' BIT TIME
2436  485A  00                          SXFER:  .BYTE   0         ;SOFTWARE 'XFER' FLAG
2437  485B  00                          SINTSTA: .BYTE  0         ;SOFTWARE COPY OF INTSTA WORD
2438  485C  00                          PADEXP: .BYTE   0         ;
2439
2440  485D  00                          ECCEXP: .BYTE   0         ;
2441  485E  00                          ECCCHR: .BYTE   0         ;ECC CHARACTER CALCULATED BY THE SOFTWARE
2442  485F  00 00                       CRCCHR: .WORD   0         ;IMAGE OF THE CRC CHARACTER
2443  4861  00                          CRCOLD: .BYTE   0         ;CRC CHAR TO USE IN PE TEST (INTERMED. STEP)
2444  4862  00                          HAVCRC: .BYTE   0         ;RECEIVED 'CRC' CHAR
2445  4863  00                          RPADCT: .BYTE   0         ;# RECEIVED PAD CHAR
2446  4864  00                          PBIT:   .BYTE   0
2447  4865  00                          CLKCRC: .BYTE   0         ;EXTRA CLOCK-TO-CRC FLAG
2448  4866  00                          TPBCNT: .BYTE   0         ;TEST PASS BYTE COUNTER
2449  4867  00                          PDCRC:  .BYTE   0         ;PAD/CRC FLAG IF SET, WMC WILL ISSUE A PAD
2450  4868  00                          LASTCRC: .BYTE  0         ;LAST CRC BYTE EXPECTED FLAG
2451  4869  00                          MOD7R:  .BYTE   0         ;IMAGE OF RESID GROUP CHAR EXPECTED
2452  486A  00                          RCHRIM: .BYTE   0         ;IMAGE OF WHAT LOADED INTO RESCHR WORD
2453  486B  00                          XFRREC: .BYTE   0         ;COUNT OF RECEIVED 'XFER' CHARACTERS
```

```
                                    .SBTTL  SUBROUTINE GENDAT
486C                                S
 (1)                                ;  ******************************************************************
                                    ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT ''SCLK'' CYCLE.
                                    ;        DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
                                    ;        'SDDRCC' FOR REFERENCE.
4860                                S
 (1)                                ;  ******************************************************************

4860   F5               12.0  GENDAT: PUSH    PSW             ;SAVE PSW + A
486D   C5               12.0          PUSH    B               ;SAVE B + C
486E   E5               12.0          PUSH    H               ;SAVE H + L
486F   3A   4F   48     13.0          LDA     DPNUM           ;GET THE LAST PATTERN # USED
4872   3C                4.0          INR     A               ;POINT TO NEXT
4873   32   4F   48     13.0          STA     DPNUM           ;SAVE NUMBER
4876   4F                4.0          MOV     C,A
4877   06   00           7.0          MVI     B,0             ;B + C HAVE OFFSET
4879   21   B1   48     10.0          LXI     H,CDATBL        ;POINT TO DATA TABLE
487C   09               10.0          DAD     B               ;ADD IN OFFSET TO H + L

487D   7E                7.0          MOV     A,M             ;GET DATA BYTE FOR DDR 'A'
487E   32   56   48     13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
4881   AF                4.0          XRA     A               ;
4882   32   57   48     13.0          STA     SDDRBT          ;SAVE DDR 'B' IMAGE
4885   32   58   48     13.0          STA     SDDRCT          ;SAVE DDR 'C' IMAGE

4888   E1               10.0  GENOUT: POP     H               ;RESTORE REGS
4889   C1               10.0          POP     B
488A   F1               10.0          POP     PSW
488B   C9               10.0          RET
```

```
                                        .SBTTL  SUBROUTINE SETDDR
                                        ;
                                        ; ***************************************************************************
                                        ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
                                        ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
                                        ;       THE MASSBUS. WMC 'LEFT' FLAG SET INDICATES THAT THIS ROUTINE IS NOT
                                        ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
                                        ;       'LEFT' FLAG IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
                                        ;       DDR FORMAT.
                                        ;
                                        ; ***************************************************************************
                                        ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR INJECTION

488C 3A 56 48        13.0  SETDDR: LDA     SDDRAT          ;GET DDRA TEMP
488F E6 3F           7.0           ANI     @77             ;STRIP 2 BITS OFF
4891 07              4.0           RLC
4892 07              4.0           RLC                     ;LEFT JUSTIFY
4893 47              4.0           MOV     B,A             ;TEMP SAVE
4894 3A 58 48        13.0          LDA     SDDRCT          ;GET DDRC TEMP
4897 E6 03           7.0           ANI     @3              ;SAVE 2 BITS
4899 B0              4.0           ORA     B
489A 32 53 48        13.0          STA     SDDRA           ;SAVE DDRA DESIRED
489D D3 D8           10.0          OUT     DDRA            ;LOAD THE REAL DDRA

489F 3A 56 48        13.0          LDA     SDDRAT          ;GET DDRA AGAIN
48A2 E6 C0           7.0           ANI     @300            ;SAVE 2 BITS
48A4 07              4.0           RLC
48A5 07              4.0           RLC
48A6 32 54 48        13.0          STA     SDDRB           ;SAVE DDRB DATA
48A9 D3 D9           10.0          OUT     DDRB            ;LOAD THE REAL DDRB

48AB 3A 55 48        13.0          LDA     SDDRC
48AE D3 DA           10.0          OUT     DDRC
48B0 C9              10.0          RET                     ;EXIT

48B1 F7                    CDATBL: .BYTE   @367            ;1ST SET OF DATA HAS KNOWN RESULTS
48B2 FB                            .BYTE   @373
48B3 00                            .BYTE   @0
48B4 23                            .BYTE   @43
48B5 88                            .BYTE   @210
48B6 C0                            .BYTE   @300
48B7 63                            .BYTE   @143            ;EXPECTED 'ECC' IS @311
                                                           ;EXPECTED INTERMEDIATE 'CRC' IS @142

48B8 12                            .BYTE   $12             ;HEX 12
48B9 23                            .BYTE   $23             ;HEX 23
48BA 34                            .BYTE   $34             ;HEX 34
48BB 45                            .BYTE   $45             ;HEX 45
48BC 56                            .BYTE   $56             ;HEX 56
48BD 67                            .BYTE   $67             ;HEX 67
48BE 78                            .BYTE   $78             ;HEX 78
48BF 89                            .BYTE   $89             ;HEX 89
48C0 9A                            .BYTE   $9A             ;HEX 9A
```

```
2536  48C1  AB                                  .BYTE   $AB             ;HEX AB
2537  48C2  BC                                  .BYTE   $BC             ;HEX BC
2538  48C3  CD                                  .BYTE   $CD             ;HEX CD
2539  48C4  DE                                  .BYTE   $DE             ;HEX DE
2540  48C5  EF                                  .BYTE   $EF             ;HEX EF
2541  48C6  F0                                  .BYTE   $F0             ;HEX F0
2542  48C7  01                                  .BYTE   $01             ;HEX 01
2543  48C8  02                                  .BYTE   $02             ;FLOATING 1'S
2544  48C9  04                                  .BYTE   $04
2545  48CA  08                                  .BYTE   $08
2546  48CB  10                                  .BYTE   $10
2547  48CC  20                                  .BYTE   $20
2548  48CD  40                                  .BYTE   $40
2549  48CE  80                                  .BYTE   $80
2550  48CF  FE                                  .BYTE   $FE             ;FLOATING 0'S
2551  48D0  FD                                  .BYTE   $FD
2552  48D1  FB                                  .BYTE   $FB
2553  48D2  F7                                  .BYTE   $F7
2554  48D3  EF                                  .BYTE   $EF
2555  48D4  DF                                  .BYTE   $DF
2556  48D5  BF                                  .BYTE   $BF
2557  48D6  7F                                  .BYTE   $7F
2558  48D7                              DTEND:                          ;END OF THE DATA TABLE
2559
2560                                    ;ROUTINE TO SHIFT CONTENTS OF ACCUMULATOR RIGHT OR LEFT EITHER 2 OR 4 TIMES
2561
2562  48D7  E6  FC          7.0   SHR4:   ANI     @374                  ;STRIP OFF 2 BITS
2563  48D9  0F              4.0           RRC                           ;SHIFT RIGHT 2 TIMES
2564  48DA  0F              4.0           RRC
2565  48DB  E6  FC          7.0   SHR2:   ANI     @374                  ;STRIP OFF 2 BITS
2566  48DD  0F              4.0           RRC                           ;SHIFT RIGHT 2 TIMES
2567  48DE  0F              4.0           RRC
2568  48DF  C9             10.0           RET
2569
2570  48E0  E6  3F          7.0   SHL4:   ANI     @77                   ;SHIFT LEFT TWICE
2571  48E2  07              4.0           RLC
2572  48E3  07              4.0           RLC
2573  48E4  E6  3F          7.0   SHL2:   ANI     @77
2574  48E6  07              4.0           RLC                           ;SHIFT LEFT TWICE
2575  48E7  07              4.0           RLC
2576  48F8  C9             10.0           RET
2577  48E9                              S
 (1)                                  ; **********************************************************************
2578                                   ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
2579                                   ;       'CLK' BIT TO A '1' THEN A '0'.
2580  48E9                              S
 (1)                                  ; **********************************************************************
2581
2582  48E9  3A  4A  48     13.0   CLKSYS: LDA     CCTLWD                ;GET SOFTWARE CLOCK CONTROL IMAGE
2583  48EC  F6  40          7.0           ORI     SSCLK                 ;ADD IN 'CLK' BIT
2584  48EE  D3  F0         10.0           OUT     CLKCTL                ;LOAD CLOCK CONTROL
2585  48F0  E6  3F          7.0           ANI     @77                   ;STRIP OFF CLOCK BIT
2586  48F2  D3  F0         10.0           OUT     CLKCTL                ;LOAD CLOCK CONTROL WORD
2587  48F4  C9             10.0           RET                           ;EXIT - CLOCK CYCLE COMPLETE
```

```
2589                                    .SBTTL  CLEAR ECC SUBROUTINE
2590  48F5                              S
 (1)                                    ; ************************************************************************
2591                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
2592                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2593                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2594  48F5                              S
 (1)                                    ; ************************************************************************
2595
2596  48F5  F5                  12.0  CLECC:  PUSH    PSW             ;SAVE THE ACCUMULATOR
2597  48F6  AF                   4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2598  48F7  32  5E  48          13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
2599  48FA  32  51  48          13.0          STA     ECCCNT          ;CLEAR THE COUNT OF ASSEMBLED BYTES
2600  48FD  F1                  10.0          POP     PSW             ;RESTORE THE ACCUMULATOR
2601  48FE  C9                  10.0          RET                     ;RETURN TO USER
```

```
                                        .SBTTL  CALCUALTE ECC CHARACTER
                                        $
                                        ; ************************************************************************
                                        ;GENECC --THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF 'ECCCHR' TO UPDATE THE CONTENTS OF 'ECCCHR' ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        $
                                        ; ************************************************************************

                                GENECC: PUSH    PSW             ;SAVE THE ACCUMULATOR
                                        PUSH    B               ;SAVE B&C
                                        PUSH    D               ;SAVE D&E
                                        PUSH    H               ;SAVE H&L
                                        LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
                                        XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
                                        MOV     E,A             ;SAVE XOR RESULT IN E
                                        ANI     $10             ;IS BIT #4 OF RESULT SET
                                        MOV     A,E             ;RESTORE XOR RESULT FROM B
                                        JZ      ECC1            ;CONTINUE IF BIT #4 RESET
                                        XRI     $23             ;ELSE-XOR WITH 23
                                ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                        XRA     A               ;CLEAR A
                                        MOV     C,A             ;CLEAR THE TRANSLATE RESULT
                                        LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
                                        CALL    TRANS           ;TRANSLATE THE BITS
                                        MOV     A,C             ;GET THE TRANSLATED RESULT
                                        STA     ECCCHR          ;STORE RESULT
                                        POP     H               ;RESTORE H&L
                                        POP     D               ;RESTORE D&E
                                        POP     B               ;RESTORE B&C
                                        POP     PSW             ;RESTORE ACCUM.
                                        RET

                                ECCTBL: $08                     ;BIT 0 = POSITION 3
                                        $20                     ;BIT 1 = POSITION 5
                                        $02                     ;BIT 2 = POSITION 1
                                        $40                     ;BIT 3 = POSITION 6
                                        $80                     ;BIT 4 = POSITION 7
                                        $01                     ;BIT 5 = POSITION 0
                                        $10                     ;BIT 6 = POSITION 4
                                        $04                     ;BIT 7 = POSITION 2
```

```
2643                                    .SBTTL  POLYNOMIAL BIT TRANSLATION
2644
2645                                    ;
2646                                    ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
2647                                    ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
2648                                    ;
2649
2650  492A  06  01            7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
2651  492C  7B                 4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
2652  492D  A0                 4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
2653  492E  CA  34  49        10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
2654  4931  79                 4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
2655  4932  B6                 7.0          ORA   M            ;'OR' IN NEW POSITION
2656  4933  4F                 4.0          MOV   C,A          ;SAVE RESULT
2657
2658  4934  78                 4.0  TRANS2: MOV   A,B
2659  4935  07                 4.0          RLC                ;POSITION MASK TO NEXT BIT
2660  4936  47                 4.0          MOV   B,A
2661  4937  D8                12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
2662  4938  23                 6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
2663  4939  C3  2C  49        10.0          JMP   TRANS1       ;PROCESS NEXT BIT
2664        0000                    .END
```

```
A      =%0007          ABCNT   484C          ADATA  = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15 = 8000          BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B         CBYTL  = 008A          CCTLWD  484A
CDATBL  48B1           CDG1H  = 0087         CDG1L  = 0086          CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094          CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028          CH0TIE= 0020          CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027          CKCRC   47CC          CKECC   475F
CKFCC   4767  G        CKECC1  47C6          CKHACK  460E          CKLOP  = 2815
CKP     4558           CKPADA  455C          CKPAD1  45C0          CKSCLK  4474
CKSC1   4483           CLCRC   482B          CLECC   48F5          CLKCNT  4852
CLKCRC  4865           CLKCTL= 00F0          CLKON   4850          CLKSYS  48E9
CLOCK   4F26           CMC0H  = 0099         CMC0L  = 0098          CMC1H  = 009B
CMC1L  = 009A          CMC2H  = 009D         CMC2L  = 009C          CMC3H  = 009F
CMC3L  = 009E          CMINH  = 0097         CMINL  = 0096          CNTCTL= 00D7
CRCCHR  485F           CRCFIN  4833          CRCOLD  4861          CRCWRD= 0018
CRC1    4817           CRC2    4822          CRC2A   47CE          CRC3    4827
CRC4    47E0           CRC5    47F5  G       CSAVE   4F9D          CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085         CTCL   = 0084          CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081         CXCTL  = 0080          CXINH  = 0083
CXINL  = 0082          C.     = 0001         C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003         C.DVA  = 0008          C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E          C.GO   = 0001          C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080         C.RCT  = 00FC          C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F          C.TAPE= 0040          C.WCS  = 00C2
D      =%0002          DATACT= 00D0          DATPE   469C          DATPEC  46A7  G
DATPEE  46AD           DATPO   4688          DATPOC  4695  G       DATPOE  469B
DBUS    4F28           DBUSCT= 00C0          DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9         DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088         DDRCTL= 00DB          DELOOP  473C
DELP1   4749           DIAFLG  4F2C          DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045         DONINT= 0010          DPNUM   484F
DSAVE   4F9E           DSE    = 0006         DTEND   48D7          D.ATH0= 0001
D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020          D.NOTW= 0040
D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080          E      =%0003
EABCNT  484D           ECCBAD= 0042          ECCCHR  485E          ECCCK   476D
ECCCK1  4782  G        ECCCK2  4788          ECCCNT  4851          ECCCOR= 0019
ECCEXP  485D           ECCOK  = 0041         ECCPE   47B4          ECCPEC  47BF  G
ECCPEE  47C5           ECCPO   47A0          ECCPOC  47AD  G       ECCPOE  47B3
ECCSTA= 001A           ECCTBL  4922          ECCTST= 000E          ECC1    4910
EDATA  = 0095          EOTCLR= 0003          ERF_G   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812         ERLPE  = 280C          ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F          E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080         E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002          E.UNC  = 0004          FIFORD= 006A
FORMAT  4F25           FORMT   484E          FWDTST= 0061          GCRID  = 0089
GCRSET= 0002           GENCRC  47FE          GENDAT  486C          GENECC  48FF
```

```
GENOUT  4888          GOODTM= 0092          H     =%0004          HAVCRC  4862
HLSAVE  4FA0          IE    = 0008          IMBYTC  484B        INTSTA= 00E0
ITERA   4F9A          I.PWR = 0020          I.RMPE= 0040        I5.5  = 0010
I6.5  = 0020          I7.5  = 0040          KCALL = 005F        KCLR  = 007B
KDEP  = 003F          KENAB = 0078          KEXAM = 003E        KEYBRD= 00C8
KEY1  = 0078          KEY10 = 006D          KEY11 = 006E        KEY12 = 006F
KEY13 = 005C          KEY14 = 005D          KEY15 = 005E        KEY16 = 005F
KEY17 = 003C          KEY18 = 003D          KEY19 = 003E        KEY2  = 0079
KEY20 = 003F          KEY3  = 007A          KEY4  = 007B        KEY5  = 0074
KEY6  = 0075          KEY7  = 0076          KEY8  = 0077        KEY9  = 006C
KINTA = 006F          KLDAD = 003D          KN0   = 003C        KN1   = 005C
KN2   = 005D          KN3   = 005E          KN4   = 006C        KN5   = 006D
KN6   = 006E          KN7   = 0074          KN8   = 0075        KN9   = 0076
KU2   = 0079          KU3   = C07A          KU8   = 0077        L     =%0005
LASTCR  4868          LBLANK= 000F          LCE   = 000B        LCH   = 000C
LCL   = 000D          LCP   = 000E          LC0   = 0000        LC1   = 0001
LC2   = 0002          LC3   = 0003          LC4   = 0004        LC5   = 0005
LC6   = 0006          LC7   = 0007          LC8   = 0008        LC9   = 0009
LDLEDA= 00CA          LDLEDB= 00CB          LDLEDC= 00CC        LDLEDD= 00CD
LDLEDE= C0CE          LDLEDF= 00CF          LKDIAG= 2800        LKKBD = 004C
LKKEY = 0049          LKLWMG= 0058          LKLWMP= 0055        LKLWPG= 0052
LKLWPP= 004F          LKMOD7= 0046          LKOPR = 0046        LPFLG   4F94
LPNUM   4F92          M     =%0006          MBSEL = 00E0        MB.A  = 0008
MB.B  = 0004          MEMTOP= 4FFF          MINUS = 000A        MM    = 8000
MOD7R   4869          MSE   = 0008          MSGN  = 0018        MTACLR= 0000
MT.ARA= 0020          MT.CPE= 0080          MT.DSE= 0001        MT.FWD= 0040
MT.INH= 0008          MT.LWR= 0004          MT.MOT= 0002        MT.NWT= 0080
MT.PEC= 0040          MT.PSB= 0004          MT.PS0= 0001        MT.PS1= 0002
MT.REV= 0020          MT.WRT= 0010          MT.Z  = 0008        M.ATA = 0080
M.CAPE= 0020          M.CONT= 0080          M.DEM = 0020        M.EBL = 0010
M.EXC = 0008          M.FAIL= 0008          M.ILR = 0010        M.INIT= 0010
M.OCC = 0020          M.ONLI= 0001          M.PE  = 0040        M.PORT= 0080
M.RDEN= 0002          M.RDPE= 0008          M.RUN = 0004        M.SCLK= 0001
M.TRA = 0040          M.UNIT= 0007          M.WCLK= 0040        M.WCLN= 0080
M.WREN= 0080          M5.5  = 0001          M6.5  = 0002        M7.5  = 0004
NOTCAP= 0088          OKAY  = 00FF          OPRRAM= 4300        OPSTRT= 0058
OPVER = 0040          PADCNT= 00D5          PADCN0  457C  G     PADCRC= 0080
PADEXP  485C          PADPE   45AE          PADPEC  45B9  G     PADPEE  45BF
PADP0   459A          PADP0C  45A7  G       PADP0E  45AD        PARITY  483C
PAR1    4845          PBIT    4864          PDCRC   4867        PDIAG = 0048
PEID  = 008A          PENAB = 004C          PESET = 0001        PEXPTB  43BF
PL    = 0081          PRDD  = 004C          PRENF = 009C        PS    = 00B2
PSTAT = 0048          PSW   =%0009          P.AMTP= 0001        P.BCTC= 0040
P.CMDP= 0020          P.INTE= 0080          P.LCS = 0040        P.LWR = 0020
P.RDP = 0002          P.RPEN= 0004          P.RPST= 0002        P.RP0E= 0020
P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010        P.SING= 0080
P.STAT= 0002          P.STPE= 0080          P.TACH= 0008        P.TUPR= 0010
P.WCSP= 0004          P.WDS = 0040          P.WFLP= 0001        P.WPEN= 0010
P.WP0E= 0008          P.WP1E= 0004          P.WP2E= 0008        P.WP3E= 0004
P.5VOK= 0002          QUE   = 281B          QUEM  = 281E        QUIT    46ED
QUIT1   4702  G       QUIT3   4717  G       QUIT4   4708        RAMT  = 0010
RARA  = 0006          RARAI = 0004          RCHBD0= 0048        RCHBD1= 0047
RCHOK = 0046          RCHRIM  486A          RCHTST= 000C        RCLRT = 000D
RCMD  = 0008          RCMLP = 0003          RCONT = 0080        RDATA = 0017
```

```
RDCLK = 0010        RDON  = 0011        READG = 0007        REND  = 0014
REQST = 2806        RESCHR= 00D1        REVTST= 0064        REWIND= 0004
RFIFOL= 0008        RGCLK = 0002        RGCRI = 0003        RIBG  = 0001
RILL  = 0012        RINST = 000C        RMCTST= 0008        RMK2  = 0013
RNOP  = 0000        RPADCT  4863        RPATH = 0001        RPBAD = 0044
RPCHI = 0001        RPCLK = 0003        RPCTL = 0009        RPEI  = 0002
RPFAIL= 0000        RPF1  = 009D        RPF2  = 009E        RPOK  = 0043
RPOSTN= 0016        RPSTA = 0015        RRCMT = 000A        RSTAT = 0002
RTIEB = 000A        RTIER = 0030        RTM   = 0005        RUNKI = 0009
RUPTST= 005E        RWDUNL= 0005        R.AMT = 0001        R.BOP = 0008
R.DATA= 0040        R.DON = 0002        R.DRDY= 0010        R.END = 0010
R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008        R.PLOD= 0008
R.PLOO= 0010        R.PLO1= 0020        R.POST= 0020        R.STNM= 0002
R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040
R.VOK = 0080        R00H  = 0081        R00L  = 0080        R01H  = 0083
R01L  = 0082        R02H  = 0085        R02L  = 0084        R03H  = 0087
R03L  = 0086        R04H  = 0089        R04L  = 0088        R05H  = 0088
R05L  = 008A        R06H  = 008D        R06L  = 008C        R07H  = 008F
R07L  = 008E        R10H  = 0091        R10L  = 0090        R11H  = 0093
R11L  = 0092        R12H  = 0095        R12L  = 0094        R13H  = 0097
R13L  = 0096        R14H  = 0099        R14L  = 0098        R15H  = 009B
R15L  = 009A        R16H  = 009D        R16L  = 009C        R17H  = 009F
R17L  = 009E        R7.5  = 0010        SCHREX  462F        SCKB1   4466
SCKGON  445F        SCKWT   4431        SCKWTC  4454  G     SCKWTE  445D
SCKWT1  4433        SCKWT2  4440        SCLKBI  4426        SDDRA   4853
SDDRAT  4856        SDDRB   4854        SDDRBT  4857        SDDRC   4855
SDDRCT  4858        SELCLR= 0000        SETATA= 00A1        SETDDR  488C
SHL2    48E4        SHL4    48E0        SHR2    48DB        SHR4    48D7
SID   = 0080        SINTST  485B        SOD   = 0080        SOE   = 0040
SP    =%0008        SSCLK = 0040        SSTEP = 0005        STACK = 4FFF
STATRM= 4F20        STPCT   4F20        STRSP = 5000        SWMCST  4859
SXFER   485A        TADR00= 0080        TADR01= 0081        TADR02= 0082
TADR03= 0083        TADR04= 0084        TADR05= 0085        TADR06= 0086
TADR07= 0087        TADR10= 0088        TADR11= 0089        TADR12= 008A
TADR13= 008B        TAMT  = 0044        TASEL = 0080        TCMD  = 0040
TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004        TC.REV= 0020
TC.WRT= 0010        TEMP    4F99        TMF   = 0099        TMRDY = 0040
TPBCNT  4866        TRANS   492A        TRANS1  492C        TRANS2  4934
TRKENA= 00D2        TSET  = 2803        TSTEND= 2818        TSTS  = 0040
TST1    4300        TST1A   43D3        TST1B   43DE        TST1BC  440B  G
TST1C   43FB        TST1D   4414        TST1DC  441D  G     TST1L   431F
TST1NB  430E        TST1ST  4305        TST1XC  466A  G     TUSEL0= 00D1
TUSEL1= 00D2        TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002
T.BOT = 0004        T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004
T.ONL = 0020        T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008
T.PS1J= 0010        T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010
T.SCLK= 0002        T1RST   4373        T1RSTA  439B        T1RSTB  439E
T1RSTC  43C7        T1RSTD  43AE        UIBG  = 00A1        VALFC   4F98
VALTB   4F95        VELTST= 005B        WDR.P = 0010        WMCCTL= 00D3
WMCERR- 00DA        WMCSTA= 00D0        WRTCLK= 0000        WRTDAT= 00D3
W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002        W.DONN= 0040
W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020        W.FMT = 0070
W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020        W.RESI= 0002
W.REV = 0004        W.ROME= 0010        W.RST = 0001        W.SKIP= 000F
```

```
W.WRIT= 0008          W.XFER= 0020          X     =%000A          XFB4C   4629  G
XFRBIT  448A          XFRB1   4491          XFRB2   44AF          XFRB2C  44C5  G
XFRB3   44CE          XFRB3A  44DC          XFRB3C  44F2  G       XFRB4   44FB
XFRB4C  4641          XFRB4D  4545          XFRB4P  4670          XFRB5   46AE
XFRB5C  46E4  G       XFRDY1  451E  G       XFRDY2  4527          XFRDY3  453C  G
XFRERR  4624          XFRGON  44A7  G       XFRRDY  4501          XFRREC  486B
XMCCK   45D0          XMCK1   45D4          XMCK1A  45D2          XMCK2   45F1
XMCK3   45F5          XMCK3C  4605  G       X.DONN= 0080          X.ENAB= 0040
X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001          Y     =%0008
.     = 493C
```

ERRORS DETECTED: 0

```
*WMC5.A78/PTP,WMC5 NLIST,PARAM,MACRO,LIST,WMC5
RUN-TIME: 5 7 0 SECONDS
CORE USED:  10K
```

TABLE OF CONTENTS

```
                                .TITLE  TUP1 - TAPE UNIT PORT TEST PART #1
                                .SBTTL  TEST  1 - LOOP CMD. TO STATUS - TU PORT #0
                                ;ID     TUP1-TAPE UNIT PORT CONTROLLER PART #1
1332  4300                      ST
 (1)                            ; ********************************  ********************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1333                            ;*LOOP COMMAND TO STATUS-TU PORT #0
1334  4300                      SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1335                            ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #0 COMMAND AND
1336                            ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
1337                            ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
1338  4300                      SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1339                            ;*BGNTST
1340                            ;*     IF TU PORT #0 NOT SELECTED BY USER
1341                            ;*     :   THEN-NEXT TEST
1342                            ;*     :   ELSE-CONTINUE
1343                            ;*     ENDIF
1344                            ;*     LOAD TU PORT #0 SELECT CODE
1345                            ;*     CALL SUBROUTINE CLEAR
1346                            ;*     CALL SUBROUTINE CONTR1
1347                            ;*ENDTST
1348  4300                      SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1349                            ;*TUP1 MICRO TEST 01
1350                            ;*TUP1 MICRO ERROR 01
1351                            ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
1352                            ;*M8955'S, M8954, M8960
1353                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1354                            ;*FATAL ERROR - TEST ABORTED
1355                            ;*
1356                            ;*TUP1 MICRO TEST 01
1357                            ;*TUP1 MICRO ERROR 14
1358                            ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
1359                            ;*M8955'S, M8954, M8960
1360                            ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
1361                            ;*ACTUAL = NNNN
1362                            ;*EXPECTED = NNNN
1363                            ;*
1364                            ;*TUP1 MICRO TEST 01
1365                            ;*TUP1 MICRO ERROR 15
1366                            ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
1367                            ;*M8955'S, M8954, M8960
1368                            ;*PTY. BIT NOT = 1 FOR DATA
1369                            ;*ACTUAL = NNNN
1370                            ;*
```

```
1371                                    ;*TUP1 MICRO TEST 01
1372                                    ;*TUP1 MICRO ERROR 16
1373                                    ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
1374                                    ;*M8955'S, M8954, M8960
1375                                    ;*PTY. BIT NOT = 0 FOR DATA
1376                                    ;*ACTUAL = NNNN
1377  4300                              S
 (1)                                    ; ***************************************************************************
1378
1379  4300                              TEST1:  TESTX   $1              ;SET UP THE TEST NUMBER
 (1)  4300  3E  C1            7.0                 MVI     A,$1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1380                                    ;%TUP1-LOOP CMD. TO STATUS-TU PORT #0
1381                                    ;&M8955'S, M8954, M8960
1382  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28       18.0                 CALL    REQST
 (1)  4308  00                                    .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                .WORD   0               ;SYSTEM '0'' COUNT
 (1)  430B  00  00                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                    .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                    .BYTE   @7              ;REQUEST CODE
1383  430F                                      RIN     R12L            ;GET THE UNITS DESIRED
 (1)  430F  DB  94           10.0                 IN      R12L                ;READ R12L INTO AC
 (1)  4311  7F                4.0                 MOV     A,A                 ;RETRY LINK
1384  4312  32  49  49        13.0               STA     UNITMP          ;STORE IN MEMORY
1385  4315  A7                 4.0               ANA     A               ;SET THE CONDITION CODE
1386  4316  C2  27  43        10.0               JNZ     TST01X          ;GO TEST THE SPECIFIED UNITS
1387                                                                     ;NO UNITS SPECIFIED
1388  4319                                      ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28       18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                 .BYTE
 (1)  431E  CD  15  28       18.0           DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  92  47       10.0                 JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1389                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1390                                    ;>FATAL ERROR - TEST ABORTED
1391  4324  C3  92  47       10.0               JMP     EXIT            ;EXIT THE TEST
1392
1393  4327  3A  49  49        13.0  TST01X: LDA     UNITMP          ;GET THE UNIT MAP
1394  432A  E6  01                7.0          ANI     @001            ;TEST UNIT 0?
1395  432C  CA  4A  43        10.0               JZ      TEST2           ;NO-CHECK FOR TEST2
1396                                                                     ;YES-RUN THE TEST
1397  432F  AF                 4.0               XRA     A               ;GET THE PORT # FOR TEST
1398  4330  CD  98  47        18.0               CALL    CLEAR           ;CLEAR ALL PORTS
1399  4333  CD  C3  47        18.0               CALL    CONTR1          ;DO THE TEST
1400
1401  4336                                      ENDTST  TST01X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4336                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4336  CD  06  28       18.0                 CALL    REQST
 (2)  4339  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (2)  433A  00  00                                .WORD                   ;SYSTEM '' COUNT
```

```
  (2)  433C  00  00                                        .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  433E  00                                            .BYTE                ;DATA COMPARE FLAG IF =1
  (2)  433F  07                                            .BYTE   7            ;REQUEST CODE
  (1)  4340  3A  9A  4F      13.0          LDA     ITERA          ;GET ITERATION COUNT
  (1)  4343  3D                4.0          DCR     A              ;DOWNCOUNT
  (1)  4344  32  9A  4F      13.0          STA     ITERA          ;SAVE COUNT
  (1)  4347  F2  27  43      10.0          JP      TST01X         ;DO TEST UNTIL TILL = 0
1402
```

```
                                .SBTTL  TEST  2 - LOOP CMD. TO STATUS - TU PORT #1
                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS-TU PORT #1
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #1 COMMAND AND
                                ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
                                ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #1 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR1
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 02
                                ;*TUP1 MICRO ERROR 14
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #1
                                ;*M8955'S, M8954, M8960
                                ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 02
                                ;*TUP1 MICRO ERROR 15
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #1
                                ;*M8955'S, M8954, M8960
                                ;*PTY. BIT NOT = 1 FOR DATA
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 02
                                ;*TUP1 MICRO ERROR 16
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #1
                                ;*M8955'S, M8954, M8960
                                ;*PTY. BIT NOT = 0 FOR DATA
                                ;*ACTUAL = NNNN
                                S
                                ; ***************************************************************************
```

```
1445  434A                                  TEST2:  TESTX   $2               ;SET UP THE TEST NUMBER
 (1)  434A  3E  02                 7.0                     MVI     A,$2            ;DEFINE THE TEST NUMBER
 (1)  434C  CD  03  28            18.0                     CALL    TSET            ;SETUP THE TEST
1446                                     ;%TUP1-LOOP CMD. TO STATUS-TU PORT #1
1447                                     ;&M8955'S, M8954, M8960
1448  434F  3A  49  49              13.0 TST02X: LDA     UNITMP          ;GET THE UNIT MAP
1449  4352  E6  02                    7.0          ANI     @002            ;TEST UNIT 1?
1450  4354  CA  73  43               10.0          JZ      TEST3           ;NO-CHECK FOR TEST3
1451  4357  3E  01                    7.0          MVI     A,@01           ;LOAD UNIT 1 SELECT CODE
1452  4359  CD  98  47              18.0          CALL    CLEAR           ;CLEAR ALL PORTS
1453  435C  CD  C3  47              18.0          CALL    CONTR1          ;TEST THIS PORT
1454  435F                                       ENDTST  TST02X
 (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  435F                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  435F  CD  06  28            18.0                     CALL    REQST
 (2)  4362  00                                             .BYTE                   ;DATA PATTERN NUMBER
 (2)  4363  00  00                                         .WORD                   ;SYSTEM "" COUNT
 (2)  4365  00  00                                         .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4367  00                                             .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4368  07                                             .BYTE   7               ;REQUEST CODE
 (1)  4369  3A  9A  4F            13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  436C  3D                     4.0          DCR     A               ;DOWNCOUNT
 (1)  436D  32  9A  4F            13.0          STA     ITERA           ;SAVE COUNT
 (1)  4370  F2  4F  43            10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1455
```

```
                                .SBTTL  TEST  3 - LOOP CMD. TO STATUS - TU PORT #2
                               ST
                               ; ***************************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*LOOP COMMAND TO STATUS-TU PORT #2
                               SD
                               ; ***************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #2 COMMAND AND
                               ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
                               ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
                               SP
                               ; ***************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*     IF TU PORT #2 NOT SELECTED BY USER
                               ;*     :   THEN-NEXT TEST
                               ;*     :   ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     LOAD TU PORT #2 SELECT CODE
                               ;*     CALL SUBROUTINE CLEAR
                               ;*     CALL SUBROUTINE CONTR1
                               ;*ENDTST
                               SE
                               ; ***************************************************************************
                               ;*ERRORS
                               ;*------
                               ;*TUP1 MICRO TEST 03
                               ;*TUP1 MICRO ERROR 14
                               ;*TUP1-LOOP CMD. TO STATUS-TU PORT #2
                               ;*M8955'S, M8954, M8960
                               ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
                               ;*ACTUAL = NNNN
                               ;*EXPECTED = NNNN
                               ;*
                               ;*TUP1 MICRO TEST 03
                               ;*TUP1 MICRO ERROR 15
                               ;*TUP1-LOOP CMD. TO STATUS-TU PORT #2
                               ;*M8955'S, M8954, M8960
                               ;*PTY. BIT NOT = 1 FOR DATA
                               ;*ACTUAL = NNNN
                               ;*
                               ;*TUP1 MICRO TEST 03
                               ;*TUP1 MICRO ERROR 16
                               ;*TUP1-LOOP CMD. TO STATUS-TU PORT #2
                               ;*M8955'S, M8954, M8960
                               ;*PTY. BIT NOT = 0 FOR DATA
                               ;*ACTUAL = NNNN
                               S
                               ; ***************************************************************************
```

```
1498  4373                                TEST3:  TESTX   $3              ;SET UP THE TEST NUMBER
 (1)  4373  3E  03              7.0                      MVI     A,$3            ;DEFINE THE TEST NUMBER
 (1)  4375  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
1499                                ;%TUP1-LOOP CMD. TO STATUS-TU PORT #2
1500                                ;@M8955'S, M8954, M8960
1501  4378  3A  49  49         13.0 TST03X: LDA     UNITMP          ;GET THE UNIT MAP
1502  437B  E6  04              7.0          ANI     @004            ;TEST FOR UNIT 2
1503  437D  CA  9C  43         10.0          JZ      TEST4           ;NO-CHECK FOR TEST4
1504  4380  3E  02              7.0          MVI     A,@02           ;GET PORT SELECT #
1505  4382  CD  98  47         18.0          CALL    CLEAR           ;CLEAR ALL PORTS
1506  4385  CD  C3  47         18.0          CALL    CONTR1          ;TEST THIS PORT
1507  4388                                    ENDTST  TST03X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4388                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4388  CD  06  28         18.0                      CALL    REQST
 (2)  438B  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  438C  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  438E  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4390  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4391  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4392  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4395  3D                  4.0          DCR     A               ;DOWNCOUNT
 (1)  4396  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  4399  F2  78  43         10.0          JP      TST03X          ;DO TEST UNTIL TILL = 0
1508
```

```
                                .SBTTL  TEST  4 - LOOP CMD. TO STATUS - TU PORT #3
439C                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP COMMAND TO STATUS-TU PORT #3
439C                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #3 COMMAND AND
                                ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
                                ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
439C                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #3 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR1
                                ;*ENDTST
439C                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 04
                                ;*TUP1 MICRO ERROR 14
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #3
                                ;*M8955'S, M8954, M8960
                                ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 04
                                ;*TUP1 MICRO ERROR 15
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #3
                                ;*M8955'S, M8954, M8960
                                ;*PTY. BIT NOT = 1 FOR DATA
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 04
                                ;*TUP1 MICRO ERROR 16
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #3
                                ;*M8955'S, M8954, M8960
                                ;*PTY. BIT NOT = 0 FOR DATA
                                ;*ACTUAL = NNNN
439C                            S
 (1)                            ; ************************************************************************
```

```
1551  439C                                    TEST4:  TESTX   $4               ;SET UP THE TEST NUMBER
 (1)  439C  3E  04                    7.0                MVI     A,$4             ;DEFINE THE TEST NUMBER
 (1)  439E  CD  03  28               18.0                CALL    TSET             ;SETUP THE TEST
1552                                        ;%TUP1-LOOP CMD. TO STATUS-TU PORT #3
1553                                        ;&M8955'S, M8954, M8960
1554  43A1  3A  49  49                13.0  TST04X: LDA     UNITMP           ;GET THE UNIT MAP
1555  43A4  E6  08                     7.0          ANI     @010             ;TEST FOR UNIT 3?
1556  43A6  CA  C5  43                10.0          JZ      TEST5            ;NO-CHECK FOR TEST5
1557  43A9  CD  98  47                18.0          CALL    CLEAR            ;CLEAN UP THE PORTS
1558  43AC  3E  03                     7.0          MVI     A,@03            ;
1559  43AE  CD  C3  47                18.0          CALL    CONTR1           ;
1560  43B1                                        ENDTST  TST04X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43B1                                          REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43B1  CD  06  28               18.0                                   CALL    REQST
 (2)  43B4  00                                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  43B5  00  00                                                          .WORD                   ;SYSTEM '#' COUNT
 (2)  43B7  00  00                                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43B9  00                                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43BA  07                                                              .BYTE   7               ;REQUEST CODE
 (1)  43BB  3A  9A  4F               13.0                LDA     ITERA            ;GET ITERATION COUNT
 (1)  43BE  3D                        4.0                DCR     A                ;DOWNCOUNT
 (1)  43BF  32  9A  4F               13.0                STA     ITERA            ;SAVE COUNT
 (1)  43C2  F2  A1  43               10.0                JP      TST04X           ;DO TEST UNTIL TILL = 0
1561
```

```
                                .SBTTL  TEST  5 - WCS PARITY INT. TEST - TU PORT #0
1564  43C5                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #0
1566  43C5                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #0 PARITY ERROR
                                ;*INTERRUPT LOGIC.
1569  43C5                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #0 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR3
                                ;*ENDTST
1579  43C5                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 21
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 22
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
                                ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 23
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 24
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
```

```
1605                                    ;*INTER. WHEN NONE EXP.
1606                                    ;*ACTUAL = NNNN
1607  43C5                              S
 (1)                                    ; ***************************************************************************
1608
1609  43C5                              TEST5:  TESTX   @05              ;INITIALIZE THE TEST
 (1)  43C5  3E  05              7.0                     MVI     A,@05           ;DEFINE THE TEST NUMBER
 (1)  43C7  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1610                                    ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
1611                                    ;&M8955'S, M8954
1612  43CA  3A  49  49          13.0  TST05X: LDA     UNITMP          ;GET THE UNIT MAP
1613  43CD  E6  01                7.0          ANI     @001            ;TEST FOR UNIT 0?
1614  43CF  CA  ED  43           10.0          JZ      TEST6
1615  43D2  AF                    4.0          XRA     A               ;GET PORT 0 SELECT CODE
1616  43D3  CD  98  47           18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
1617  43D6  CD  84  48           18.0          CALL    CONTR3          ;ISSUE THE TEST
1618  43D9                                    ENDTST  TST05X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43D9                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43D9  CD  06  28         18.0                     CALL    REQST
 (2)  43DC  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  43DD  00  00                                      .WORD                   ;SYSTEM '@' COUNT
 (2)  43DF  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43E1  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43E2  07                                          .BYTE   7               ;REQUEST CODE
 (1)  43E3  3A  9A  4F          13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  43E6  3D                   4.0          DCR     A               ;DOWNCOUNT
 (1)  43E7  32  9A  4F          13.0          STA     ITERA           ;SAVE COUNT
 (1)  43EA  F2  CA  43          10.0          JP      TST05X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST  6 - WCS PARITY INT. - TU PORT #1
1621  43ED                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #1
1623  43ED                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #1 PARITY ERROR
                                ;*INTERRUPT LOGIC.
1626  43ED                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #1 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR3
                                ;*ENDTST
1636  43ED                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 21
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 22
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
                                ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 23
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 24
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
```

```
1662                                    ;*INTER. WHEN NONE EXP.
1663                                    ;*ACTUAL = NNNN
1664  43ED                              S
 (1)                                    ; ********************************************************************************
1665
1666  43ED                              TEST6:  TESTX   @06             ;INITIALIZE THE TEST
 (1)  43ED  3E  06              7.0                      MVI     A,@06           ;DEFINE THE TEST NUMBER
 (1)  43EF  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
1667                                    ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
1668                                    ;@M8955'S, M8954
1669  43F2  3A  49  49          13.0  TST06X: LDA     UNITMP          ;GET THE UNIT MAP
1670  43F5  E6  02                7.0          ANI     @002            ;TEST FOR UNIT 1?
1671  43F7  CA  16  44           10.0          JZ      TEST7
1672  43FA  3E  01                7.0          MVI     A,1             ;GET PORT #
1673  43FC  CD  98  47           18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
1674  43FF  CD  84  48           18.0          CALL    CONTR3          ;DO THE TEST
1675  4402                                    ENDTST  TST06X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4402                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4402  CD  06  28         18.0                      CALL    REQST
 (2)  4405  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4406  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  4408  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  440A  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  440B  07                                           .BYTE   7               ;REQUEST CODE
 (1)  440C  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  440F  3D                  4.0          DCR     A               ;DOWNCOUNT
 (1)  4410  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  4413  F2  F2  43         10.0          JP      TST06X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST  7 - WCS PARITY INT. - TU PORT #2
1678  4416                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #2
1680  4416                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #2 PARITY ERROR
                                ;*INTERRUPT LOGIC.
1683  4416                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #2 SELECT CODE #
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR3
                                ;*ENDTST
1693  4416                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 21
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 22
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*M8955'S, M8954
                                ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 23
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 24
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*INTER. WHEN NONE EXP.
```

```
1719                                    ;*ACTUAL = NNNN
1720  4416                              $
 (1)                                    ; ***************************************************************************
1721
1722  4416                              TEST7:  TESTX   @07              ;INITIALIZE THE TEST
 (1)  4416  3E  07              7.0                      MVI     A,@07           ;DEFINE THE TEST NUMBER
 (1)  4418  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
1723                                    ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
1724                                    ;&M8955'S, M8954
1725  441B  3A  49  49         13.0    TST07X: LDA     UNITMP          ;GET THE UNIT MAP
1726  441E  E6  04               7.0           ANI     @004            ;TEST FOR UNIT 2?
1727  4420  CA  3F  44         10.0           JZ      TEST10
1728  4423  3E  02               7.0           MVI     A,2             ;GET PORT #
1729  4425  CD  98  47         18.0           CALL    CLEAR           ;CLEAN UP THE PORTS
1730  4428  CD  84  48         18.0           CALL    CONTR3          ;DO THE TEST
1731  442B                                   ENDTST  TST07X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  442B                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  442B  CD  06  28         18.0                      CALL    REQST
 (2)  442E  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  442F  00  00                                       .WORD                   ;SYSTEM '#' COUNT
 (2)  4431  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4433  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4434  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4435  3A  9A  4F         13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  4438  3D                  4.0           DCR     A               ;DOWNCOUNT
 (1)  4439  32  9A  4F         13.0           STA     ITERA           ;SAVE COUNT
 (1)  443C  F2  1B  44         10.0           JP      TST07X          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 10 - WCS PARITY INT. - TU PORT #3
1734  443F                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1735                            ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #3
1736  443F                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1737                            ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #3 PARITY ERROR
1738                            ;*INTERRUPT LOGIC.
1739  443F                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1740                            ;*BGNTST
1741                            ;*     IF TU PORT #3 NOT SELECTED BY USER
1742                            ;*     :   THEN-NEXT TEST
1743                            ;*     :   ELSE-CONTINUE
1744                            ;*     ENDIF
1745                            ;*     LOAD TU PORT #3 SELECT CODE
1746                            ;*     CALL SUBROUTINE CLEAR
1747                            ;*     CALL SUBROUTINE CONTR3
1748                            ;*ENDTST
1749  443F                      SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1750                            ;*TUP1 MICRO TEST 10
1751                            ;*TUP1 MICRO ERROR 21
1752                            ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
1753                            ;*M8955'S, M8954
1754                            ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
1755                            ;*ACTUAL = NNNN
1756                            ;*
1757                            ;*TUP1 MICRO TEST 10
1758                            ;*TUP1 MICRO ERROR 22
1759                            ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
1760                            ;*M8955'S, M8954
1761                            ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
1762                            ;*ACTUAL = NNNN
1763                            ;*
1764                            ;*TUP1 MICRO TEST 10
1765                            ;*TUP1 MICRO ERROR 23
1766                            ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
1767                            ;*M8955'S, M8954
1768                            ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
1769                            ;*ACTUAL = NNNN
1770                            ;*
1771                            ;*TUP1 MICRO TEST 10
1772                            ;*TUP1 MICRO ERROR 24
1773                            ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
1774                            ;*M8955'S, M8954
```

```
1775                                    ;*INTER. WHEN NONE EXP.
1776                                    ;*ACTUAL = NNNN
1777  443F                              $
 (1)                                    ; ************************************************************************
1778
1779  443F                              TEST10: TESTX   @010            ;INITIALIZE THE TEST
 (1)  443F  3E  08            7.0                    MVI     A,@010      ;DEFINE THE TEST NUMBER
 (1)  4441  CD  03  28       18.0                    CALL    TSET        ;SETUP THE TEST
1780                                    ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
1781                                    ;@M8955'S, M8954
1782  4444  3A  49  49        13.0     TST10X: LDA     UNITMP          ;GET THE UNIT MAP
1783  4447  E6  08             7.0             ANI     @010            ;TEST FOR UNIT 3?
1784  4449  CA  68  44        10.0             JZ      TEST11
1785  444C  3E  03             7.0             MVI     A,3             ;GET THE PORT #
1786  444E  CD  98  47        18.0             CALL    CLEAR           ;CLEAN UP THE PORTS
1787  4451  CD  84  48        18.0             CALL    CONTR3          ;DO THE TEST
1788  4454                                    ENDTST  TST10X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4454                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4454  CD  06  28       18.0                    CALL    REQST
 (2)  4457  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4458  00  00                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  445A  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  445C  00                                      .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  445D  07                                      .BYTE   7               ;REQUEST CODE
 (1)  445E  3A  9A  4F        13.0             LDA     ITERA           ;GET ITERATION COUNT
 (1)  4461  3D                 4.0             DCR     A               ;DOWNCOUNT
 (1)  4462  32  9A  4F        13.0             STA     ITERA           ;SAVE COUNT
 (1)  4465  F2  44  44        10.0             JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 11 - LOOP AMTIE TO AMTIE - TU PORT #0
                                        ST
                                        ; ***************************************************************************
                                        ;*TEST TITLE
                                        ;*----- -----
                                        ;*LOOP AMTIE TO AMTIE TEST-TU PORT #0
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
                                        ;*#0 AMTIE REGISTER.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*     IF TU PORT #0 NOT SELECTED BY USER
                                        ;*     :   THEN-NEXT TEST
                                        ;*     :   ELSE-CONTINUE
                                        ;*     ENDIF
                                        ;*     CALL SUBROUTINE CLEAR
                                        ;*     CALL SUBROUTINE CONTR2
                                        ;*ENDTST
                                        SE
                                        ; ***************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*TUP1 MICRO TEST 11
                                        ;*TUP1 MICRO ERROR 17
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #0
                                        ;*M8955'S, M8954
                                        ;*DATA WRITTEN TO AMTIE NOT = DATA READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 11
                                        ;*TUP1 MICRO ERROR 20
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #0
                                        ;*M8955'S, M8954
                                        ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        S
                                        ; ***************************************************************************

                                        TEST11: TESTX   @11             ;SET UP THE TEST NUMBER
 (1) 4468 3E 09          7.0                            MVI     A,@11           ;DEFINE THE TEST NUMBER
 (1) 446A CD 03 28      18.0                            CALL    TSET            ;SETUP THE TEST
                                        ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #0
                                        ;&M8955'S, M8954
 4468 3A 49 49          13.0  TST11X: LDA     UNITMP          ;GET THE UNIT MAP  .
 446D E6 01              7.0           ANI     @001            ;TEST UNIT 0?
 4470 CA 91 44          10.0           JZ      TEST12          ;NO-CHECK FOR TEST12
```

```
1829                                                                    ;YES-RUN THE TEST
1830  4475  3E  00          7.0          MVI     A,@00
1831  4477  CD  98  47      18.0         CALL    CLEAR                  ;CLEAN UP THE PORTS
1832  447A  CD  27  48      18.0         CALL    CONTR2                 ;
1833  447D                               ENDTST  TST11X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  447D                          REQ     7                          ;FAKE CALL TO KEEP TEST ALIVE
 (2)  447D  CD  06  28      18.0                                CALL    REQST
 (2)  4480  00                                                  .BYTE                  ;DATA PATTERN NUMBER
 (2)  4481  00  00                                              .WORD                  ;SYSTEM '@' COUNT
 (2)  4483  00  00                                              .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4485  00                                                  .BYTE                  ;DATA COMPARE FLAG IF =1
 (2)  4486  07                                                  .BYTE   7              ;REQUEST CODE
 (1)  4487  3A  9A  4F      13.0         LDA     ITERA                  ;GET ITERATION COUNT
 (1)  448A  3D                4.0       DCR     A                      ;DOWNCOUNT
 (1)  448B  32  9A  4F      13.0         STA     ITERA                  ;SAVE COUNT
 (1)  448E  F2  6D  44      10.0         JP      TST11X                 ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 12 - LOOP AMTIE TO AMTIE - TU PORT #1
                                        ST
                                        ; ***************************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*LOOP AMTIE TO AMTIE TEST-TU PORT #1
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
                                        ;*#1 AMTIE REGISTER.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*    IF TU PORT #1 NOT SELECTED BY USER
                                        ;*    :   THEN-NEXT TEST
                                        ;*    :   ELSE-CONTINUE
                                        ;*    ENDIF
                                        ;*    CALL SUBROUTINE CLEAR
                                        ;*    CALL SUBROUTINE CONTR2
                                        ;*ENDTST
                                        SE
                                        ; ***************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*TUP1 MICRO TEST 12
                                        ;*TUP1 MICRO ERROR 17
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #1
                                        ;*M8955'S, M8954
                                        ;*DATA WRITTEN TO AMTIE NOT = DATA READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 12
                                        ;*TUP1 MICRO ERROR 20
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #1
                                        ;*M8955'S, M8954
                                        ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        S
                                        ; ***************************************************************************

                                        TEST12: TESTX   @12             ;SET UP THE TEST NUMBER
 (1)  4491  3E  0A          7.0                  MVI     A,@12           ;DEFINE THE TEST NUMBER
 (1)  4493  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #1
                                        ;&M8955'S, M8954
 4496  3A  49  49          13.0  TST12X: LDA     UNITMP          ;GET THE UNIT MAP
 4499  E6  02               7.0         ANI     @002            ;TEST UNIT 1?
 449B  CA  BA  44          10.0         JZ      TEST13          ;NO-CHECK FOR TEST13
```

```
1874  449E  3E  01        7.0        MVI     A,@01          ;
1875  44A0  CD  98  47    18.0       CALL    CLEAR          ;CLEAN UP PORTS
1876  44A3  CD  27  48    18.0       CALL    CONTR2         ;
1877  44A6                          ENDTST  TST12X
 (1,                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44A6                          REQ     7              ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44A6  CD  06  28    18.0                        CALL    REQST
 (2)  44A9  00                                        .BYTE                  ;DATA PATTERN NUMBER
 (2)  44AA  00  00                                    .WORD                  ;SYSTEM '@' COUNT
 (2)  44AC  00  00                                    .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44AE  00                                        .BYTE                  ;DATA COMPARE FLAG IF =1
 (2)  44AF  07                                        .BYTE  7               ;REQUEST CODE
 (1)  44B0  3A  9A  4F    13.0       LDA     ITERA          ;GET ITERATION COUNT
 (1)  44B3  3D        4.0        DCR     A              ;DOWNCOUNT
 (1)  44B4  32  9A  4F    13.0       STA     ITERA          ;SAVE COUNT
 (1)  44B7  F2  96  44    10.0       JP      TST12X         ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 13 - LOOP AMTIE TO AMTIE - TU PORT #2
44BA                                    ST
 (1)                                    ; ************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*LOOP AMTIE TO AMTIE TEST-TU PORT #2
44BA                                    SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
                                        ;*#2 AMTIE REGISTER.
44BA                                    SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*     IF TU PORT #2 NOT SELECTED BY USER
                                        ;*     :   THEN-NEXT TEST
                                        ;*     :   ELSE-CONTINUE
                                        ;*     ENDIF
                                        ;*     CALL SUBROUTINE CLEAR
                                        ;*     CALL SUBROUTINE CONTR2
                                        ;*ENDTST
44BA                                    SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*TUP1 MICRO TEST 13
                                        ;*TUP1 MICRO ERROR 17
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #2
                                        ;*M8955'S, M8954
                                        ;*DATA WRITTEN TO AMTIE NOT = DATA READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 13
                                        ;*TUP1 MICRO ERROR 20
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #2
                                        ;*M8955'S, M8954
                                        ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
44BA                                    S
 (1)                                    ; ************************************************************************
                                        
44BA                                    TEST13: TESTX   @13             ;SET UP THE TEST NUMBER
 (1) 44BA  3E  0B          7.0                          MVI     A,@13           ;DEFINE THE TEST NUMBER
 (1) 44BC  CD  03  28     18.0                          CALL    TSET            ;SETUP THE TEST
                                        ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #2
                                        ;&M8955'S, M8954
44BF  3A  49  49          13.0  TST13X: LDA     UNITMP          ;GET THE UNIT MAP
44C2  E6  04               7.0        ANI     @004            ;TEST FOR UNIT 2
44C4  CA  E3  44          10.0        JZ      TEST14          ;NO-CHECK FOR TEST14
```

```
 1918  44C7  3E  02        7.0          MVI     A,@02           ;
 1919  44C9  CD  98  47    18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
 1920  44CC  CD  27  48    18.0         CALL    CONTR2          ;
 1921  44CF                             ENDTST  TST13X
  (1)                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  44CF                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  44CF  CD  06  28    18.0                                 CALL    REQST
  (2)  44D2  00                                                 .BYTE                   ;DATA PATTERN NUMBER
  (2)  44D3  00  00                                             .WORD                   ;SYSTEM '@' COUNT
  (2)  44D5  00  00                                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  44D7  00                                                 .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  44D8  07                                                 .BYTE   7               ;REQUEST CODE
  (1)  44D9  3A  9A  4F    13.0         LDA     ITERA           ;GET ITERATION COUNT
  (1)  44DC  3D             4.0         DCR     A               ;DOWNCOUNT
  (1)  44DD  32  9A  4F    13.0         STA     ITERA           ;SAVE COUNT
  (1)  44E0  F2  BF  44    10.0         JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
1923                                    .SBTTL  TEST 14 - LOOP AMTIE TO AMTIE - TU PORT #3
1924  44E3                              ST
 (1)                                    ;*****************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1925                                    ;*LOOP AMTIE TO AMTIE TEST-TU PORT #3
1926  44E3                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1927                                    ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
1928                                    ;*#3 AMTIE REGISTER.
1929  44E3                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1930                                    ;*BGNTST
1931                                    ;*     IF TU PORT #3 NOT SELECTED BY USER
1932                                    ;*     :   THEN-NEXT TEST
1933                                    ;*     :   ELSE-CONTINUE
1934                                    ;*     ENDIF
1935                                    ;*     CALL SUBROUTINE CLEAR
1936                                    ;*     CALL SUBROUTINE CONTR2
1937                                    ;*ENDTST
1938  44E3                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1939                                    ;*TUP1 MICRO TEST 14
1940                                    ;*TUP1 MICRO ERROR 17
1941                                    ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #3
1942                                    ;*M8955'S, M8954
1943                                    ;*DATA WRITTEN TO AMTIE NOT = DATA READ
1944                                    ;*ACTUAL = NNNN
1945                                    ;*EXPECTED = NNNN
1946                                    ;*
1947                                    ;*TUP1 MICRO TEST 14
1948                                    ;*TUP1 MICRO ERROR 20
1949                                    ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #3
1950                                    ;*M8955'S, M8954
1951                                    ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
1952                                    ;*ACTUAL = NNNN
1953                                    ;*EXPECTED = NNNN
1954  44E3                              S
 (1)                                    ;*****************************************************************************
1955
1956  44E3                              TEST14: TESTX   @014              ;SET UP THE TEST NUMBER
 (1)  44E3  3E  0C            7.0                    MVI     A,@014          ;DEFINE THE TEST NUMBER
 (1)  44E5  CD  03  28       18.0                    CALL    TSET            ;SETUP THE TEST
1957                                    ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #3
1958                                    ;&M8955'S, M8954
1959  44E8  3A  49  49         13.0     TST14X: LDA     UNITMP          ;GET THE UNIT MAP
1960  44EB  E6  08               7.0             ANI     @010            ;TEST FOR UNIT 3?
1961  44ED  CA  0C  45          10.0             JZ      TEST15          ;NO-CHECK FOR TEST15
```

```
1962  44F0  3E  03        7.0        MVI     A,@03
1963  44F2  CD  98  47   18.0        CALL    CLEAR          ;CLEAN UP THE PORT;
1964  44F5  CD  27  48   18.0        CALL    CONTR2
1965  44F8                           ENDTST  TST14X
 (1)                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44F8                         REQ     7              ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44F8  CD  06  28   18.0                        CALL    REQST
 (2)  44FB  00                                      .BYTE                  ;DATA PATTERN NUMBER
 (2)  44FC  00  00                                  .WORD                  ;SYSTEM '@' COUNT
 (2)  44FE  00  00                                  .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4500  00                                      .BYTE                  ;DATA COMPARE FLAG IF -1
 (2)  4501  07                                      .BYTE   7              ;REQUEST CODE
 (1)  4502  3A  9A  4F   13.0        LDA     ITERA          ;GET ITERATION COUNT
 (1)  4505  3D            4.0        DCR     A              ;DOWNCOUNT
 (1)  4506  32  9A  4F   13.0        STA     ITERA          ;SAVE COUNT
 (1)  4509  F2  E8  44   10.0        JP      TST14X         ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 15 - LOOP 'W/R ENA'' TO 'W/R ENA'' - TU PORT #0
1968  450C                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #0
1970  450C                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH BITS TO BE
                                ;*SET AND RESET.
1973  450C                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #0 NOT SELECTED BY USER
                                ;*     :    THEN-NEXT TEST
                                ;*     :    ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT TU PORT 0
                                ;*     SET THE WRITE PATH ENABLE BITS FOR TU PORT #0
                                ;*     INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #0
                                ;*     IF WRITE PATH ENABLE BIT SET
                                ;*     :    THEN-CONTINUE
                                ;*     :    ELSE-ERROR
                                ;*     ENDIF
                                ;*     RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #0
                                ;*     INPUT THE READ PATH ENABLE BIT FOR TU PORT #0
                                ;*     IF READ PATH ENABLE BITS RESET
                                ;*     :    THEN-CONTINUE
                                ;*     :    ELSE-ERROR
                                ;*     ENDIF
                                ;*ENDTST
1994  450C                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 15
                                ;*TUP1 MICRO ERROR 02
                                ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #0
                                ;*M8955'S
                                ;*EXPECTED PORT 0 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 15
                                ;*TUP1 MICRO ERROR 03
                                ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #0
                                ;*M8955'S
                                ;*EXPECTED PORT 0 WRITE PATH ENAB CLEARED & READ PATH ENAB SET
                                ;*ACTUAL = NNNN
```

```
2009                                    ;*EXPECTED = NNNN
2010  450C                              S
 (1)                                    ; ***************************************************************************
2011
2012  450C                              TEST15: TESTX   @15              ;INITIALIZE THE TEST
 (1)  450C  3E  0D            7.0                     MVI     A,@15       ;DEFINE THE TEST NUMBER
 (1)  450E  CD  03  28       18.0                     CALL    TSET        ;SETUP THE TEST
2013                                    ;%TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #0
2014                                    ;&M8955'S
2015
2016  4511  3A  49  49        13.0  TST15X: LDA     UNITMP              ;GET THE UNIT MAP
2017  4514  E6  01             7.0          ANI     @01                 ;TEST FOR UNIT 0
2018  4516  CA  7A  45         10.0          JZ      TEST16              ;NO-GO CHECK FOR TEST 16
2019                                                                    ;
2020  4519  CD  98  47        18.0          CALL    CLEAR               ;CLEAN UP THE PORTS
2021  451C  DB  E0             10.0          IN      INTSTA
2022  451E  E6  80              7.0          ANI     BIT7
2023  4520  C6  00              7.0          ADI     0
2024  4522  D3  E0             10.0          OUT     MBSEL               ;SELECT PORT 0
2025  4524  3E  10              7.0  DPEN00: MVI     A,P.WPEN            ;LOAD THE WRITE DATA PATH ENABLE BIT
2026  4526  D3  4C             10.0          OUT     PENAB               ;SET THE DATA PATH ENABLE BITS
2027  4528  00                  4.0          NOP
2028  4529  DB  D1             10.0          IN      TUSEL0              ;GET THE ENABLE STATUS WORD
2029  452B  E6  3F              7.0          ANI     $3F                 ;STRIP OUT THE DUAL TU PORT BIT
2030  452D  FE  08              7.0          CPI     P.WPOEN!0           ;SEE IF PROPER BITS SET
2031  452F  CA  3F  45         10.0          JZ      DPEN01              ;JUMP IF OK
2032  4532                                  ROUT    ADATA
 (1)  4532  D3  94            10.0                  OUT     ADATA       ;WRITE AC INTO ADATA
 (1)  4534  7F                 4.0                  MOV     A,A         ;RETRY LINK
2033  4535  3E  08              7.0          MVI     A,P.WPOEN!0
2034  4537                                  ROUT    EDATA
 (1)  4537  D3  95            10.0                  OUT     EDATA       ;WRITE AC INTO EDATA
 (1)  4539  7F                 4.0                  MOV     A,A         ;RETRY LINK
2035  453A                                  ERRB    DPEN00,DPEN01
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  453A  CD  12  28        18.0                  CALL    ERLPB       ;PROCESS ERROR - DO 2.3
 (1)        0002                                    MSGN    =       MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453D  02                                      .BYTE   MSGN        ;MESSAGE NUMBER ID
 (1)  453E  00                                      .BYTE               ;PRINT ROUTINE NUMBER
 (1)  453F  CD  15  28        18.0  DPEN01::        CALL    CKLOP       ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4542  DA  24  45        10.0                  JC      DPEN00      ;LOOP ADDRESS IF LOOP SPECIFIED
2036                            ;>EXPECTED PORT 0 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2037  4545  3E  04              7.0  DPEN02: MVI     A,P.RPEN            ;CLEAR WRITE PATH ENABLE & SET READ PATH ENABLE
2038  4547  D3  4C             10.0          OUT     PENAB               ;CLEAR THE DATA PATH ENABLE BITS
2039  4549  00                  4.0          NOP
2040  454A  DB  D1             10.0          IN      TUSEL0              ;GET THE ENABLE STATUS WORD
2041  454C  E6  3F              7.0          ANI     $3F                 ;STRIP OUT THE DUAL TU PORT BIT
2042  454E  FE  20              7.0          CPI     P.RPOEN!0           ;SEE IF PROPER BITS CLEAR
2043  4550  CA  60  45         10.0          JZ      DPEN03              ;JUMP IF OK
2044  4553                                  ROUT    ADATA
 (1)  4553  D3  94            10.0                  OUT     ADATA       ;WRITE AC INTO ADATA
 (1)  4555  7F                 4.0                  MOV     A,A         ;RETRY LINK
2045  4556  3E  20              7.0          MVI     A,P.RPOEN!0
2046  4558                                  ROUT    EDATA
```

```
   (1)  4558  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
   (1)  455A  7F               4.0                MOV     A,A             ;RETRY LINK
  2047  455B                                 ERRB    DPEN02,DPEN03
   (1)                                       ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
   (1)  455B  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
   (1)        0003                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
   (1)  455E  03                              .BYTE   MSGN                    ;MESSAGE NUMBER ID
   (1)  455F  00                              .BYTE                           ;PRINT ROUTINE NUMBER
   (1)  4560  CD  15  28      18.0            DPEN03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
   (1)  4563  DA  45  45      10.0                    JC      DPEN02          ;LOOP ADDRESS IF LOOP SPECIFIED
  2048                                ;>EXPECTED PORT 0 WRITE PATH ENAB CLEARED & READ PATH ENAB SET
  2049  4566                                 ENDTST  TST15X
   (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
   (2)  4566                                 REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
   (2)  4566  CD  06  28      18.0                    CALL    REQST
   (2)  4569  00                              .BYTE                           ;DATA PATTERN NUMBER
   (2)  456A  00  00                          .WORD                           ;SYSTEM '*' COUNT
   (2)  456C  00  00                          .WORD                           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
   (2)  456E  00                              .BYTE                           ;DATA COMPARE FLAG IF =1
   (2)  456F  07                              .BYTE   7                       ;REQUEST CODE
   (1)  4570  3A  9A  4F      13.0                    LDA     ITERA           ;GET ITERATION COUNT
   (1)  4573  3D               4.0                    DCR     A               ;DOWNCOUNT
   (1)  4574  32  9A  4F      13.0                    STA     ITERA           ;SAVE COUNT
   (1)  4577  F2  11  45      10.0                    JP      TST15X          ;DO TEST UNTIL TILL = 0
```

```
2051                                .SBTTL  TEST 16 - LOOP 'W/R ENA'' TO 'W/R ENA'' - TU PORT #1
2052  457A                         ST
 (1)                                ;*********************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
2053                                ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #1
2054  457A                         SD
 (1)                                ;*********************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
2055                                ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH BITS TO BE
2056                                ;*SET AND RESET.
2057  457A                         SP
 (1)                                ;*********************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
2058                                ;*BGNTST
2059                                ;*     IF TU PORT #1 NOT SELECTED BY USER
2060                                ;*     :   THEN-NEXT TEST
2061                                ;*     :   ELSE-CONTINUE
2062                                ;*     ENDIF
2063                                ;*     CALL SUBROUTINE CLEAR
2064                                ;*     SELECT TU PORT #1
2065                                ;*     SET THE WRITE PATH ENABLE BITS FOR TU PORT #1
2066                                ;*     INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #1
2067                                ;*     IF WRITE PATH ENABLE BIT SET
2068                                ;*     :   THEN-CONTINUE
2069                                ;*     :   ELSE-ERROR
2070                                ;*     ENDIF
2071                                ;*     RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #1
2072                                ;*     INPUT THE READ PATH ENABLE BIT FOR TU PORT #1
2073                                ;*     IF READ PATH ENABLE BITS RESET
2074                                ;*     :   THEN-CONTINUE
2075                                ;*     :   ELSE-ERROR
2076                                ;*     ENDIF
2077                                ;*ENDTST
2078  457A                         SE
 (1)                                ;*********************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
2079                                ;*TUP1 MICRO TEST 16
2080                                ;*TUP1 MICRO ERROR 04
2081                                ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #1
2082                                ;*M8955'S
2083                                ;*EXPECTED PORT 1 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2084                                ;*ACTUAL = NNNN
2085                                ;*EXPECTED = NNNN
2086                                ;*
2087                                ;*TUP1 MICRO TEST 16
2088                                ;*TUP1 MICRO ERROR 05
2089                                ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #1
2090                                ;*M8955'S
2091                                ;*EXPECTED PORT 1 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
2092                                ;*ACTUAL = NNNN
```

```
                                        ;*EXPECTED = NNNN
                                        $
                                        ; ************************************************************************************

                                        TEST16: TESTX   @16             ;INITIALIZE THE TEST
                                                        MVI     A,@16           ;DEFINE THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #1
                                        ;&M8955'S

                                        TST16X: LDA     UNITMP          ;GET THE UNIT MAP
                                                ANI     @02             ;TEST FOR UNIT 1
                                                JZ      TEST17          ;NO-GO CHECK FOR TEST 17
                                                                        ;
                                                CALL    CLEAR           ;CLEAN UP THE PORTS
                                                IN      INTSTA
                                                ANI     BIT7
                                                ADI     1
                                                OUT     MBSEL           ;SELECT PORT 1
                                        DPEN10: MVI     A,P.WPEN        ;LOAD THE DATA PATH ENABLE BITS
                                                OUT     PENAB           ;SET THE DATA PATH ENABLE BITS
                                                NOP
                                                IN      TUSEL0          ;GET THE ENABLE STATUS WORD
                                                ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
                                                CPI     P.WP1EN!1       ;SEE IF PROPER BITS SET
                                                JZ      DPEN11          ;JUMP IF OK
                                                ROUT    ADATA
                                                        OUT     ADATA           ;WRITE AC INTO ADATA
                                                        MOV     A,A             ;RETRY LINK
                                                MVI     A,P.WP1EN!1
                                                ROUT    EDATA
                                                        OUT     EDATA           ;WRITE AC INTO EDATA
                                                        MOV     A,A             ;RETRY LINK
                                                ERRB    DPEN10,DPEN11
                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                DPEN11::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      DPEN10          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>EXPECTED PORT 1 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                        DPEN12: MVI     A,P.RPEN        ;CLEAR WRITE PATH ENAB. & SET REAR PATH
                                                OUT     PENAB           ;CLEAR THE DATA PATH ENABLE BITS
                                                NOP
                                                IN      TUSEL0          ;GET THE ENABLE STATUS WORD
                                                ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
                                                CPI     P.RP1EN!1       ;SEE IF PROPER BITS CLEAR
                                                JZ      DPEN13          ;JUMP IF OK
                                                ROUT    ADATA
                                                        OUT     ADATA           ;WRITE AC INTO ADATA
                                                        MOV     A,A             ;RETRY LINK
                                                MVI     A,P.RP1EN!1
                                                ROUT    EDATA
```

```
  (1)  45C6  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  45C8  7F               4.0                MOV     A,A             ;RETRY LINK
 2131  45C9                                ERRB    DPEN12,DPEN13
  (1)                                     ;FLAG ERROR - WITH ACTUAL DATA 'ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  45C9  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        0005                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  45CC  05                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  45CD  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  45CE  CD  15  28      18.0        DPEN13::                CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  45D1  DA  B3  45      10.0                    JC      DPEN12          ;LOOP ADDRESS IF LOOP SPECIFIED
 2132                                ;>EXPECTED PORT 1 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
 2133  45D4                                ENDTST  TST16X
  (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  45D4                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  45D4  CD  06  28      18.0                    CALL    REQST
  (2)  45D7  00                                      .BYTE                   ;DATA PATTERN NUMBER
  (2)  45D8  00  00                                  .WORD                   ;SYSTEM '@' COUNT
  (2)  45DA  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  45DC  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  45DD  07                                      .BYTE   7               ;REQUEST CODE
  (1)  45DE  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
  (1)  45E1  3D               4.0                DCR     A               ;DOWNCOUNT
  (1)  45E2  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
  (1)  45E5  F2  7F  45      10.0                JP      TST16X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 17 - LOOP 'W/R ENA'' TO 'W/R ENA'' - TU PORT #2
45E8                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #2
45E8                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH BITS TO BE
                                ;*SET AND RESET.
45E8                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #2 NOT SELECTED BY USER
                                ;*     :    THEN-NEXT TEST
                                ;*     :    ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT TU PORT #2
                                ;*     INPUT THE SINGLE TU PORT BIT
                                ;*     IF SINGLE PORT SET
                                ;*     :    THEN-ERROR
                                ;*     :    ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     SET THE WRITE PATH ENABLE BITS FOR TU PORT #2
                                ;*     INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #2
                                ;*     IF WRITE PATH ENABLE BIT SET
                                ;*     :    THEN-CONTINUE
                                ;*     :    ELSE-ERROR
                                ;*     ENDIF
                                ;*     RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #2
                                ;*     INPUT THE READ PATH ENABLE BITS FOR TU PORT #2
                                ;*     IF READ PATH ENABLE BIT RESET
                                ;*     :    THEN-CONTINUE
                                ;*     :    ELSE-ERROR
                                ;*     ENDIF
                                ;*ENDTST
45E8                            SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 17
                                ;*TUP1 MICRO ERROR 06
                                ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA'' - TU PORT 2
                                ;*M8955'S
                                ;*'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
                                ;*BIT 7 (217(8)) MUST BE =0 TO USE PORT 2
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 17
```

```
                                        ;*TUP1 MICRO ERROR 07
                                        ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA'' - TU PORT 2
                                        ;*M8955'S
                                        ;*EXPECTED PORT 2 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 17
                                        ;*TUP1 MICRO ERROR 10
                                        ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA'' - TU PORT 2
                                        ;*M8955'S
                                        ;*EXPECTED PORT 2 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN

45E8                                    S
 (1)                                    ; ************************************************************************************

                                        TEST17: TESTX   @17             ;INITIALIZE THE TEST
 (1) 45E8  3E  0F           7.0,                        MVI     A,@17       ;DEFINE THE TEST NUMBER
 (1) 45EA  CD  03  28      18.0                         CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #2
                                        ;&M8955'S

45ED  3A  49  49          13.0  TST17X: LDA     UNITMP          ;GET THE UNIT MAP
45F0  E6  04               7.0        ANI     @4              ;TEST FOR UNIT 2
45F2  CA  6B  46          10.0        JZ      TEST20          ;NO-GO CHECK FOR TEST 20
45F5  CD  98  47          18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
45F8  DB  E0              10.0        IN      INTSTA
45FA  E6  80               7.0        ANI     BIT7
45FC  C6  02               7.0        ADI     2
45FE  D3  E0              10.0        OUT     MBSEL           ;SELECT PORT 2
4600  DB  D1              10.0        IN      TUSEL0          ;GET THE SELECT STATUS WORD 0
4602                                   ROUT    ADATA           ;SAVE IN CASE OF ERROR
 (1) 4602  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4604  7F               4.0                MOV     A,A             ;RETRY LINK
4605  E6  80               7.0        ANI     P.SINGLE        ;SAVE THE SINGLE PORT STATUS BIT
4607  CA  0F  46          10.0        JZ      TST17P          ;JUMP IF BIT IS CLEARED (MULTI PORT)
460A                                   ERRA    TST17X,TST17P
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1) 460A  CD  0F  28      18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0006                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 460D  06                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 460E  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1) 460F  CD  15  28      18.0        TST17P::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4612  DA  ED  45      10.0                JC      TST17X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                   ;>'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
                                   ;>BIT 7 (200(8)) MUST BE =0 TO USE PORT 2
4615  3E  10               7.0  DPEN20: MVI     A,P.WPEN        ;LOAD THE DATA PATH ENABLE BITS
4617  D3  4C              10.0        OUT     PENAB           ;SET THE DATA PATH ENABLE BITS
4619  00                   4.0        NOP
461A  DB  D2              10.0        IN      TUSEL1          ;GET THE ENABLE STATUS WORD
461C  E6  3F               7.0        ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
461E  FE  0A               7.0        CPI     P.WP2EN!2       ;SEE IF PROPER BITS SET
```

```
2219  4620  CA 30 46   10.0          JZ      DPEN21          ;JUMP IF OK
2220  4623                          POUT    ADATA
 (1)  4623  D3 94       10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4625  7F          4.0                  MOV     A,A             ;RETRY LINK
2221  4626  3E 0A       7.0          MVI     A,P.WP2EN!2
2222  4628                          ROUT    EDATA
 (1)  4628  D3 95       10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  462A  7F          4.0                  MOV     A,A             ;RETRY LINK
2223  462B                          ERRB    DPEN20,DPEN21
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  462B  CD 12 28   18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0007                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  462E  07                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  462F  00                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4630  CD 15 28   18.0          DPEN21::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4633  DA 15 46   10.0                 JC      DPEN20          ;LOOP ADDRESS IF LOOP SPECIFIED
2224                          ;>EXPECTED PORT 2 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2225  4636  3E 04       7.0  DPEN22: MVI     A,P.RPEN        ;SET READ PATH ENABLE
2226  4638  D3 4C       10.0         OUT     PENAB           ;CLEAR THE DATA PATH ENABLE BITS
2227  463A  00          4.0          NOP
2228  463B  DB D2       10.0         IN      TUSEL1          ;GET THE ENABLE STATUS WORD
2229  463D  E6 3F       7.0          ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
2230  463F  FE 22       7.0          CPI     P.RP2EN!2       ;SEE IF PROPER BITS CLEAR
2231  4641  CA 51 46   10.0          JZ      DPEN23          ;JUMP IF OK
2232  4644                          ROUT    ADATA
 (1)  4644  D3 94       10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4646  7F          4.0                  MOV     A,A             ;RETRY LINK
2233  4647  3E 22       7.0          MVI     A,P.RP2EN!2
2234  4649                          ROUT    EDATA
 (1)  4649  D3 95       10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  464B  7F          4.0                  MOV     A,A             ;RETRY LINK
2235  464C                          ERRB    DPEN22,DPEN23
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  464C  CD 12 28   18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0008                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  464F  08                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4650  00                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4651  CD 15 28   18.0          DPEN23::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4654  DA 36 46   10.0                 JC      DPEN22          ;LOOP ADDRESS IF LOOP SPECIFIED
2236                          ;>EXPECTED PORT 2 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
2237  4657                          ENDTST  TST17X
 (1)                              ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4657                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4657  CD 06 28   18.0                 CALL    REQST
 (2)  465A  00                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  465B  00 00                           .WORD                   ;SYSTEM '*' COUNT
 (2)  465D  00 00                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  465F  00                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4660  07                              .BYTE   7               ;REQUEST CODE
 (1)  4661  3A 9A 4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4664  3D          4.0          DCR     A               ;DOWNCOUNT
 (1)  4665  32 9A 4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  4668  F2 ED 45   10.0          JP      TST17X          ;DO TEST UNTIL TILL = 0
```

```
2239                                    .SBTTL  TEST 20 - LOOP 'W/R ENA'' TO 'W/R ENA'' - TU PORT #3
2240  466B                              ST
 (1)                                    ;*****************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
2241                                    ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #3
2242  466B                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2243                                    ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH ENABLE
2244                                    ;*BITS TO BE SET AND RESET.
2245  466B                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2246                                    ;*BGNTST
2247                                    ;*     IF TU PORT #3 NOT SELECTED BY USER
2248                                    ;*     :   THEN-NEXT TEST
2249                                    ;*     :   ELSE-CONTINUE
2250                                    ;*     ENDIF
2251                                    ;*     CALL SUBROUTINE CLEAR
2252                                    ;*     SELECT TU PORT #3
2253                                    ;*     INPUT THE SINGLE TU PORT BIT
2254                                    ;*     IF SINBLE PORT SET
2255                                    ;*     :   THEN-ERROR
2256                                    ;*     :   ELSE-CONTINUE
2257                                    ;*     ENDIF
2258                                    ;*     SET THE WRITE PATH ENABLE BITS FOR TU PORT #3
2259                                    ;*     INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #3
2260                                    ;*     IF WRITE PATH ENABLE BIT SET
2261                                    ;*     :   THEN-CONTINUE
2262                                    ;*     :   ELSE-ERROR
2263                                    ;*     ENDIF
2264                                    ;*     RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #3
2265                                    ;*     INPUT THE READ PATH ENABLE BIT FOR TU PORT #3
2266                                    ;*         IF READ PATH ENABLE BITS RESET
2267                                    ;*         :   THEN-CONTINUE
2268                                    ;*         :   ELSE-ERROR
2269                                    ;*         ENDIF
2270                                    ;*ENDTST
2271  466B                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2272                                    ;*TUP1 MICRO TEST 20
2273                                    ;*TUP1 MICRO ERROR 11
2274                                    ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #3
2275                                    ;*M8955'S
2276                                    ;*'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
2277                                    ;*BIT 7 (220(8)) MUST BE =0 TO USE PORT 3
2278                                    ;*
2279                                    ;*TUP1 MICRO TEST 20
2280                                    ;*TUP1 MICRO ERROR 12
```

```
                                        ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #3
                                        ;*M8955'S
                                        ;*EXPECTED PORT 3 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 20
                                        ;*TUP1 MICRO ERROR 13
                                        ;*TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #3
                                        ;*M8955'S
                                        ;*EXPECTED PORT 3 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
466B                                    S
 (1)                                    ; ************************************************************************************
                                        TEST20: TESTX   @20             ;INITIALIZE THE TEST
 (1) 466B 3E 10          7.0                            MVI     A,@20           ;DEFINE THE TEST NUMBER
 (1) 466D CD 03 28      18.0                            CALL    TSET            ;SETUP THE THE TEST
                                        ;%TUP1-LOOP 'W/R ENA'' TO 'W/R ENA''-TU PORT #3
                                        ;&M8955'S

4670 3A 49 49          13.0 TST20X: LDA     UNITMP             ;GET THE UNIT MAP
4673 E6 08              7.0          ANI     @10                ;TEST FOR UNIT 3
4675 CA EE 46          10.0          JZ      TEST21             ;NO-GO CHECK FOR TEST 21
4678 CD 98 47          18.0          CALL    CLEAR              ;CLEAN UP THE PORTS
467B DB E0             10.0          IN      INTSTA
467D E6 80              7.0          ANI     BIT7
467F C6 03              7.0          ADI     3
4681 D3 E0             10.0          OUT     MBSEL              ;SELECT PORT 3
4683 DB D1             10.0          IN      TUSEL0             ;GET THE SELECT STATUS WORD 0
4685                                 ROUT    ADATA              ;SAVE IN CASE OF ERROR
 (1) 4685 D3 94         10.0                  OUT     ADATA              ;WRITE AC INTO ADATA
 (1) 4687 7F             4.0                  MOV     A,A                ;RETRY LINK
4688 E6 80              7.0          ANI     P.SINGLE           ;SAVE THE SINGLE PORT STATUS BIT
468A CA 92 46          10.0          JZ      TST20P             ;JUMP IF BIT IS CLEARED (MULTI PORT)
468D                                 ERR     TST20X,TST20P
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 468D CD 09 28      18.0                  CALL    ERLP               ;PROCESS ERROR - DO 2.3
 (1)      0009                         MSGN    =       MSGN+1             ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4690 09                           .BYTE   MSGN               ;MESSAGE NUMBER ID
 (1) 4691 00                           .BYTE
 (1) 4692 CD 15 28      18.0          TST20P::        CALL    CKLOP              ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4695 DA 70 46      10.0                  JC      TST20X             ;LOOP ADDRESS IF LOOP SPECIFIED
                                 ;>'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
                                 ;>BIT 7 (200(8)) MUST BE =0 TO USE PORT 3
4698 3E 10              7.0 DPEN30: MVI     A,P.WPEN           ;LOAD THE DATA PATH ENABLE BITS
469A D3 4C             10.0          OUT     PENAB              ;SET THE DATA PATH ENABLE BITS
469C 00                 4.0          NOP
469D DB D2             10.0          IN      TUSEL1             ;GET THE ENABLE STATUS WORD
469F E6 3F              7.0          ANI     $3F                ;STRIP OUT THE DUAL TU PORT BIT
46A1 FE 07              7.0          CPI     P.WP3EN!3          ;SEE IF PROPER BITS SET
46A3 CA B3 46          10.0          JZ      DPEN31             ;JUMP IF OK
46A6                                 ROUT    ADATA
```

```
  (1)  46A6  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  46A8  7F               4.0                 MOV     A,A             ;RETRY LINK
2323   46A9  3E  07           7.0         MVI     A,P.WP3EN!3
2324   46AB                           ROUT    EDATA
  (1)  46AB  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  46AD  7F               4.0                 MOV     A,A             ;RETRY LINK
2325   46AE                           ERRB    DPEN30,DPEN31
  (1)                              ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
  (1)  46AE  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        000A                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46B1  0A                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46B2  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  46B3  CD  15  28      18.0         DPEN31::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46B6  DA  98  46      10.0                 JC      DPEN30          ;LOOP ADDRESS IF LOOP SPECIFIED
2326                       ;>EXPECTED PORT 3 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2327   46B9  3E  04           7.0 DPEN32: MVI     A,P.RPEN        ;SET READ PATH ENABLE
2328   46BB  D3  4C          10.0         OUT     PENAB           ;CLEAR THE DATA PATH ENABLE BITS
2329   46BD  00               4.0         NOP
2330   46BE  DB  D2          10.0         IN      TUSEL1          ;GET THE ENABLE STATUS WORD
2331   46C0  E6  3F           7.0         ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
2332   46C2  FE  13           7.0         CPI     P.RP3EN!3       ;SEE IF PROPER BITS CLEAR
2333   46C4  CA  D4  46       10.0         JZ      DPEN33          ;JUMP IF OK
2334   46C7                           ROUT    ADATA
  (1)  46C7  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  46C9  7F               4.0                 MOV     A,A             ;RETRY LINK
2335   46CA  3E  13           7.0         MVI     A,P.RP3EN!3
2336   46CC                           ROUT    EDATA
  (1)  46CC  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  46CE  7F               4.0                 MOV     A,A             ;RETRY LINK
2337   46CF                           ERRB    DPEN32,DPEN33
  (1)                              ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
  (1)  46CF  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        000B                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46D2  0B                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46D3  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  46D4  CD  15  28      18.0         DPEN33::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46D7  DA  B9  46      10.0                 JC      DPEN32          ;LOOP ADDRESS IF LOOP SPECIFIED
2338                       ;>EXPECTED PORT 3 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
2339   46DA                           ENDTST  TST20X
  (1)                              ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  46DA                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  46DA  CD  06  28      18.0                 CALL    REQST
  (2)  46DD  00                                 .BYTE                   ;DATA PATTERN NUMBER
  (2)  46DE  00  00                             .WORD                   ;SYSTEM '*' COUNT
  (2)  46E0  00  00                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  46E2  00                                 .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  46E3  07                                 .BYTE   7               ;REQUEST CODE
  (1)  46E4  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
  (1)  46E7  3D               4.0         DCR     A               ;DOWNCOUNT
  (1)  46E8  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
  (1)  46EB  F2  70  46      10.0         JP      TST20X          ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 21 - LOOP ''TACH'' TO ''TACH'' - TU PORT #0
2342  46EE                              ST
                                        ; ***************************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*LOOP TACH TO TACH TEST - TU PORT 0
2344  46EE                              SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
2346  46EE                              SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*      IF TU PORT 0 NOT SELECTED BY USER
                                        ;*      :   THEN-NEXT TEST
                                        ;*      :   ELSE-CONTINUE
                                        ;*      ENDIF
                                        ;*      CALL SUBROUTINE CLEAR
                                        ;*      GET PORT 0 SELECT CODE
                                        ;*      CALL SUBROUTINE CONTR4
                                        ;*ENDTST
2356  46EE                              SE
                                        ; ***************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*TUP1 MICRO TEST 21
                                        ;*TUP1 MICRO ERROR 25
                                        ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #0
                                        ;*M8955'S
                                        ;*''TACH'' DID NOT SET
                                        ;*
                                        ;*TUP1 MICRO TEST 21
                                        ;*TUP1 MICRO ERROR 26
                                        ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #0
                                        ;*M8955'S
                                        ;*''TACH'' DID NOT RESET
2368  46EE                              S
                                        ; ***************************************************************************

2370  46EE                              TEST21: TESTX   @21             ;INITIALIZE THE TEST
      46EE  3E  11              7.0                 MVI     A,@21       ;DEFINE THE TEST NUMBER
      46F0  CD  03  28         18.0                 CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #0
                                        ;&M8955'S
2373  46F3  3A  49  49         13.0     TST21X: LDA     UNITMP          ;GET THE UNIT MAP
2374  46F6  E6  01               7.0             ANI     @01             ;TEST FOR UNIT 0
2375  46F8  CA  17  47         10.0             JZ      TEST22          ;NO-GO CHECK FOR TEST 22
                                                                         ;
2377  46FB  CD  98  47         18.0             CALL    CLEAR           ;CLEAN UP THE PORTS
2378  46FE  3E  00               7.0             MVI     A,0             ;GET PORT 0 CODE
2379  4700  CD  12  49         18.0             CALL    CONTR4          ;TEST THIS PORT
```

```
2380  4703                                        ENDTST  TST21X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4703                              REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4703  CD  06  28      18.0                         CALL    REQST
 (2)  4706  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4707  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  4709  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  470B  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  470C  07                                           .3YTE   7               ;REQUEST CODE
 (1)  470D  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  4710  3D               4.0         DCR     A               ;DOWNCOUNT
 (1)  4711  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
 (1)  4714  F2  F3  46      10.0         JP      TST21X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 22 - LOOP ''TACH'' TO ''TACH'' - TU PORT #1
4717                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP TACH TO TACH TEST-TU PORT #1
4717                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
4717                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT #1 NOT SELECTED BY USER
                                ;*      :   THEN-NEXT TEST
                                ;*      :   ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE CLEAR
                                ;*      GET PORT 1 SELECT CODE
                                ;*      CALL SUBROUTINE CONTR4
                                ;*ENDTST
4717                            SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 22
                                ;*TUP1 MICRO ERROR 25
                                ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #1
                                ;*M8955'S
                                ;*''TACH'' DID NOT SET
                                ;*
                                ;*TUP1 MICRO TEST 22
                                ;*TUP1 MICRO ERROR 26
                                ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #1
                                ;*M8955'S
                                ;*''TACH'' DID NOT RESET
4717                            S
                                ;*****************************************************************************

4717                            TEST22: TESTX   @22              ;INITIALIZE THE TEST
4717  3E  12          7.0                       MVI     A,@22          ;DEFINE THE TEST NUMBER
4719  CD  03  28     18.0                       CALL    TSET           ;SETUP THE TEST
                                ;%TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #1
                                ;&M8955'S
471C  3A  49  49       13.0     TST22X: LDA     UNITMP         ;GET THE UNIT MAP
471F  E6  02            7.0             ANI     @02            ;TEST FOR UNIT 1
4721  CA  40  47        10.0             JZ      TEST23         ;NO-GO CHECK FOR TEST 23
                                        ;-
4724  CD  98  47        18.0             CALL    CLEAR          ;CLEAN UP THE PORTS
4727  3E  01            7.0             MVI     A,1            ;GET PORT 1 SELECT CODE
4729  CD  12  49        18.0             CALL    CONTR4         ;DO THE TEST
```

```
2421  472C                                          ENDTST  TST22X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  472C                                         REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  472C  CD  06  28        18.0                                 CALL    REQST
 (2)  472F  00                                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  4730  00  00                                                 .WORD                   ;SYSTEM '*' COUNT
 (2)  4732  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4734  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4735  07                                                     .BYTE   7               ;REQUEST CODE
 (1)  4736  3A  9A  4F        13.0                 LDA     ITERA                ;GET ITERATION COUNT
 (1)  4739  3D                 4.0                 DCR     A                    ;DOWNCOUNT
 (1)  473A  32  9A  4F        13.0                 STA     ITERA                ;SAVE COUNT
 (1)  473D  F2  1C  47        10.0                 JP      TST22X               ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 23 - LOOP ''TACH'' TO ''TACH'' - TU PORT #2
4740                                    ST
 (1)                                    ; ************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*LOOP TACH TO TACH TEST-TU PORT #2
4740                                    SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
4740                                    SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*      IF TU PORT #2 NOT SELECTED BY USER
                                        ;*      :   THEN-NEXT TEST
                                        ;*      :   ELSE-CONTINUE
                                        ;*      ENDIF
                                        ;*      CALL SUBROUTINE CLEAR
                                        ;*      GET PORT 2 SELECT CODE
                                        ;*      CALL SUBROUTINE CONTR4
                                        ;*ENDTST
4740                                    SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*TUP1 MICRO TEST 23
                                        ;*TUP1 MICRO ERROR 25
                                        ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #2
                                        ;*M8955'S
                                        ;*''TACH'' DID NOT SET
                                        ;*
                                        ;*TUP1 MICRO TEST 23
                                        ;*TUP1 MICRO ERROR 26
                                        ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #2
                                        ;*M8955'S
                                        ;*''TACH'' DID NOT RESET
4740                                    S
 (1)                                    ; ************************************************************************

4740                                    TEST23: TESTX   @23             ;INITIALIZE THE TEST
 (1) 4740  3E  13          7.0                          MVI     A,@23    ;DEFINE THE TEST NUMBER
 (1) 4742  CD  03  28     18.0                          CALL    TSET     ;SETUP THE TEST
                                        ;%TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #2
                                        ;&M8955'S
4745  3A  49  49          13.0  TST23X: LDA     UNITMP          ;GET THE UNIT MAP
4748  E6  04                7.0        ANI     @04             ;TEST FOR UNIT 2
474A  CA  69  47           10.0        JZ      TEST24          ;NO-GO CHECK FOR TEST 24
                                                               ;
474D  CD  98  47           18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
4750  3E  02                7.0        MVI     A,2             ;GET PORT 2 SELECT CODE
4752  CD  12  49           18.0        CALL    CONTR4          ;DO THE TEST
```

```
                                                ENDTST  TST23X
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 2462  4755                                                     CALL    REQST
  (1)                                                           .BYTE                   ;DATA PATTERN NUMBER
  (2)  4755                                                     .WORD                   ;SYSTEM '*' COUNT
  (2)  4755  CD  06  28        18.0                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4758  00                                               .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4759  00  00                                           .BYTE   7               ;REQUEST CODE
  (2)  475B  00  00
  (2)  475D  00
  (2)  475E  07
  (1)  475F  3A  9A  4F        13.0           LDA     ITERA           ;GET ITERATION COUNT
  (1)  4762  3D                 4.0           DCR     A               ;DOWNCOUNT
  (1)  4763  32  9A  4F        13.0           STA     ITERA           ;SAVE COUNT
  (1)  4766  F2  45  47        10.0           JP      TST23X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 24 - LOOP 'TACH' TO 'TACH' - TU PORT #3
 4769                           ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP TACH TO TACH TEST-TU PORT #3
 4769                           SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
 4769                           SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    GET PORT 3 SELECT CODE
                                ;*    CALL SUBROUTINE CONTR4
                                ;*ENDTST
 4769                           SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 24
                                ;*TUP1 MICRO ERROR 25
                                ;*TUP1-LOOP 'TACH' TO 'TACH'-TU PORT #3
                                ;*M8955'S
                                ;*'TACH' DID NOT SET
                                ;*
                                ;*TUP1 MICRO TEST 24
                                ;*TUP1 MICRO ERROR 26
                                ;*TUP1-LOOP 'TACH' TO 'TACH'-TU PORT #3
                                ;*M8955'S
                                ;*'TACH' DID NOT RESET
 4769                           S
                                ; ***************************************************************************

 4769                           TEST24: TESTX   @24             ;INITIALIZE THE TEST
 4769  3E   14          7.0                 MVI     A,@24       ;DEFINE THE TEST NUMBER
 4768  CD   03   28     18.0                CALL    TSET        ;SETUP THE TEST
                                ;%TUP1-LOOP 'TACH' TO 'TACH'-TU PORT #3
                                ;&M8955'S
 476E  3A   49   49     13.0    TST24X: LDA     UNITMP      ;GET THE UNIT MAP
 4771  E6   08          7.0            ANI     @010        ;TEST FOR UNIT 3
 4773  CA   92   47     10.0            JZ      EXIT        ;EXIT THE TEST IF NOT SET
                                        ;
 4776  CD   98   47     18.0            CALL    CLEAR       ;CLEAN UP THE PORTS
 4779  3E   03          7.0            MVI     A,3         ;GET PORT 3 SELECT CODE
 477B  CD   12   49     18.0            CALL    CONTR4      ;DO THE TEST
```

```
2503  477E                                        ENDTST  TST24X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  477E                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  477E  CD  06  28        18.0                                  CALL    REQST
 (2)  4781  00                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4782  00  00                                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  4784  00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4786  00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4787  07                                                      .BYTE   7               ;REQUEST CODE
 (1)  4788  3A  9A  4F        13.0                  LDA     ITERA           ;GET ITERATION COUNT
 (1)  478B  3D                 4.0                  DCR     A               ;DOWNCOUNT
 (1)  478C  32  9A  4F        13.0                  STA     ITERA           ;SAVE COUNT
 (1)  478F  F2  6E  47        10.0                  JP      TST24X          ;DO TEST UNTIL TILL = 0
2504
2505  4792  CD  98  47        18.0  EXIT:           CALL    CLEAR           ;LEAVE THE PORTS CLEAN
2506  4795  C3  18  28        10.0                  JMP     TSTEND          ;END OF TEST
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*********************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*********************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*********************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************************
4798  F5                        12.0    CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4799  C5                        12.0            PUSH    B
479A  06  00                     7.0             MVI     B,0             ;START TO CLEAR AT PORT #0
479C  DB  E0                    10.0    CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
479E  E6  80                     7.0             ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
47A0  B0                         4.0             ORA     B               ;ADD IN THE SELECTED PORT #
47A1  D3  E0                    10.0             OUT     MBSEL           ;RESET TO THIS PORT
47A3  3E  80                     7.0             MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
47A5  D3  40                    10.0             OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
47A7  AF                         4.0             XRA     A               ;CLEAR TU COMMAND A
47A8  D3  40                    10.0             OUT     TCMD            ;
47AA  3E  81                     7.0             MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
47AC  D3  40                    10.0             OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
47AE  3E  00                     7.0             MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
47B0  D3  40                    10.0             OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
47B2  AF                         4.0             XRA     A
47B3  D3  44                    10.0             OUT     TAMT            ;CLEAR AMTIE WORD
47B5  D3  48                    10.0             OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
47B7  D3  4C                    10.0             OUT     PENAB           ;CLEAR PORT ENABLE WORD
47B9  04                         4.0             INR     B               ;POINT TO THE NEXT PORT TO CLEAR
47BA  78                         4.0             MOV     A,B
47BB  FE  04                     7.0             CPI     4               ;DONE?
47BD  C2  9C  47                10.0             JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
47C0  C1                        10.0             POP     B               ;RESET B & C
47C1  F1                        10.0             POP     PSW             ;ALL DONE
47C2  C9                        10.0             RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE CONTR1 - LOOP COMMAND TO STATUS TESTS 1 - 4
                                $SUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*SUBROUTINE CONTROL 1 - LOOP COMMAND TO STATUS
                                $D
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE WILL ESTABLISH THE LOOP WRITE/READ CONDITION ON THE
                                ;*SPECIFIED PORT AND TRANSFER COMMANDS (0-377(8)) CHECKING EACH ONE FOR
                                ;*PROPER TRANSFER TO THE STATUS REGISTER.
                                $P
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*     CLEAR THE COMMAND DATA
                                ;*     BGNDO
                                ;*     :    ISSUE THE CONTENTS OF COMMAND DATA TO TU PORT COMMAND BYTE
                                ;*     :    WAIT
                                ;*     :    INPUT THE TU PORT STATUS BYTE
                                ;*     :         IF STATUS=COMMAND WRITTEN
                                ;*     :         :    THEN-CONTINUE
                                ;*     :         :    ELSE-ERROR
                                ;*     :         ENDIF
                                ;*     :    CALCULATE PARITY ON THE CONTENTS OF COMMAND DATA
                                ;*     :         IF CALCULATED PARTY=THE TU PORT STATUS PARITY
                                ;*     :         :    THEN-CONTINUE
                                ;*     :         :    ELSE-ERROR
                                ;*     :         ENDIF
                                ;*     :    INCREMENT THE TU PORT COMMAND DATA
                                ;*     :    DO UNTIL THE TU PORT COMMAND DATA=ZERO
                                ;*     ENDDO
                                ;*ENDSUB
                                $
                                ; ***************************************************************************
 47C3  47              4.0  CONTR1: MOV     B,A
 47C4  DB  E0         10.0         IN      INTSTA          ;GET THE SELECT DATA
 47C6  E6  80          7.0         ANI     BIT7            ;SAVE JUST THE MASSBUS SELECT BIT
 47C8  B0              4.0         ORA     B               ;ADD IN THE SELECTED PORT #
 47C9  D3  E0         10.0         OUT     MBSEL           ;SELECT IT
 47CB  3E  40          7.0         MVI     A,P.LCS         ;LOAD THE LOOP CMD. TO STAT. CMD BIT
 47CD  D3  48         10.0         OUT     PDIAG           ;SET LOOP MODE ON THE PORT
 47CF  21  4A  49     10.0         LXI     H,DATA          ;LOAD ADDRESS OF DATA
 47D2  AF              4.0         XRA     A
 47D3  77              7.0         MOV     M,A             ;STORE IN DATA
 47D4  7E              7.0  LP01:    MOV     A,M             ;GET THE DATA
 47D5  47              4.0         MOV     B,A             ;COPY TO B
 47D6  D3  40         10.0         OUT     TCMD            ;STORE IN THE CMD. REGISTER
 47D8  00              4.0         NOP                     ;
 47D9  DB  40         10.0         IN      TSR             ;READ THE STATUS
 47DB  B8              4.0         CMP     B               ;COMPARE ACTUAL AND EXP.
```

```
2596  47DC  CA  EB  47   10.0          JZ      LP02            ;EQUAL - CONTINUE
2597  47DF                            ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  47DF  D3  94          10.0               OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47E1  7F               4.0               MOV     A,A             ;RETRY LINK
2598  47E2  7E               7.0       MOV     A,M             ;GET EXP. DATA
2599  47E3                            ROUT    EDATA           ;STORE EXP. DATA
 (1)  47E3  D3  95          10.0               OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47E5  7F               4.0               MOV     A,A             ;RETRY LINK
2600  47E6                            ERRB    LP01,LP02,0
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  47E6  CD  12  28      18.0               CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000C                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47E9  0C                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47EA  00                           .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  47EB  CD  15  28      18.0       LP02::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47EE  DA  D4  47      10.0               JC      LP01            ;LOOP ADDRESS IF LOOP SPECIFIED
2601                        ;>DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
2602  47F1  7E               7.0       MOV     A,M             ;GET THE DATA
2603  47F2                            ROUT    ADATA           ;STORE THE DATA IN CAS
 (1)  47F2  D3  94          10.0               OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47F4  7F               4.0               MOV     A,A             ;RETRY LINK
2604  47F5  A7               4.0       ANA     A               ;SET THE CONDITION CODE
2605  47F6  E2  0E  48       10.0       JPO     UNIT00          ;ODD NUMBER OF ONES
2606  47F9  DB  48           10.0       IN      PSTAT           ;ELSE - EVEN NUMBER OF ONES
2607  47FB  E6  04            7.0       ANI     P.WCSP          ;PTY. BIT SHOULD BE SET
2608  47FD  C2  05  48       10.0       JNZ     LP03            ;PTY. CORRECT - CONTINUE TEST
2609  4800                            ERRA    LP01,LP03
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4800  CD  0F  28      18.0               CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      000D                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4803  0D                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4804  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4805  CD  15  28      18.0       LP03::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4808  DA  D4  47      10.0               JC      LP01            ;LOOP ADDRESS IF LOOP SPECIFIED
2610                        ;>PTY. BIT NOT = 1 FOR DATA
2611  480B  C3  20  48       10.0       JMP     UNIT0X          ;CONTINUE TEST
2612                                    ;
2613  480E  DB  48           10.0  UNIT00: IN      PSTAT           ;PTY. BIT SHOULD BE RESET
2614  4810  E6  04            7.0       ANI     P.WCSP          ;
2615  4812  CA  1A  48       10.0       JZ      LP04            ;CONTINUE IF CORRECT
2616  4815                            ERRA    LP01,LP04
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4815  CD  0F  28      18.0               CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      000E                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4818  0E                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4819  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  481A  CD  15  28      18.0       LP04::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  481D  DA  D4  47      10.0               JC      LP01            ;LOOP ADDRESS IF LOOP SPECIFIED
2617                        ;>PTY. BIT NOT = 0 FOR DATA
2618  4820  7E               7.0  UNIT0X: MOV     A,M             ;GET DATA
2619  4821  3C               4.0       INR     A               ;UPDATE THE DATA BYTE
2620  4822  77               7.0       MOV     M,A             ;STORE THE UPDATED DATA
2621  4823  C2  D4  47       10.0       JNZ     LP01            ;CONTINUE UNTIL DONE
2622  4826  C9               10.0       RET
```

```
                                        .SBTTL  SUBROUTINE CONTR2 - LOOP AMTIE TO AMTIE TESTS 11 - 14
                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*SUBROUTINE CONTROL 2 - LOOP AMTIE TO AMTIE
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE WILL TRANSFER DATA (0-377(8)) ACROSS THE TU PORT AMTIE
                                        ;*LINES AS WELL AS TEST THE AMTIE PARITY BIT (BY USING THE LSB OF THE
                                        ;*DATA AS AMTIE PARITY).
                                        SP
                                        ;*********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     BGNDO
                                        ;*     :    ISSUE THE CONTENTS OF AMTIE DATA TO TU PORT #0 AMTIE DATA REGISTER
                                        ;*     :    ISSUE THE LSB OF THE AMTIE DATA REGISTER TO TU PORT AMTIE PARITY REGISTER
                                        ;*     :    INPUT THE AMTIE DATA REGISTER
                                        ;*     :         IF AMTIE DATA READ=AMTIE DATA
                                        ;*     :         :    THEN-CONTINUE
                                        ;*     :         :    ELSE-ERROR
                                        ;*     :         ENDIF
                                        ;*     :    INPUT THE AMTIE PARITY BIT
                                        ;*     :         IF THE AMTIE PARITY READ=AMTIE PARITY WRITTEN
                                        ;*     :         :    THEN-CONTINUE
                                        ;*     :         :    ELSE-ERROR
                                        ;*     :         ENDIF
                                        ;*     :    CLEAR TU PORT LOOP W/R BIT
                                        ;*     :    INCREMENT THE AMTIE DATA
                                        ;*     :    DO UNTIL THE AMTIE DATA=ZERO
                                        ;*     ENDDO
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************
 4827  47                      4.0      CONTR2: MOV     B,A             ;
 4828  DB   E0                10.0              IN      INTSTA          ;GET THE MASSBUS SELECT DATA
 482A  E6   80                 7.0              ANI     BIT7            ;SAVE JUST THE SELECT BIT
 482C  B0                      4.0              ORA     B               ;ADD IN THE SELECTED PORT #
 482D  D3   E0                10.0              OUT     MBSEL           ;SELECT PORT TO TEST
 482F  3E   20                 7.0              MVI     A,P.LWR         ;CLEAR PORT CONTROL BYTE
 4831  D3   48                10.0              OUT     PDIAG           ;SET LOOP MODE ON THE PORT
 4833  21   4A   49           10.0              LXI     H,DATA          ;LOAD ADDRESS OF DATA
                                                                        ;
 4836  AF                     4.0              XRA     A               ;CLEAR THE ACCUMULATOR
 4837  77                     7.0              MOV     M,A             ;STORE IN DATA
 4838  7E                     7.0      ALP01:  MOV     A,M             ;GET THE DATA
 4839  47                     4.0              MOV     B,A             ;COPY TO B
 483A  D3   44                10.0              OUT     TAMT            ;STORE IN THE AMTIE REGISTER
 483C  E6   01                 7.0              ANI     $01             ;MASK OFF THE LSB
 483E  F6   20                 7.0              ORI     P.LWR           ;OR IN THE LOOP WRITE TO READ BIT
```

```
2668  4840  D3  48        10.0        OUT     PDIAG           ;WRITE THE AMTIE PTY.
2669  4842  00             4.0        NOP                     ;
2670  4843  DB  44        10.0        IN      TAMT            ;READ THE AMTIE REGISTER
2671                                                          ;
2672  4845  B8             4.0        CMP     B               ;COMPARE ACTUAL AND EXP.
2673  4846  CA  55  48     10.0        JZ      ALP02           ;EQUAL - CONTINUE
2674  4849                              ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4849  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  484B  7F             4.0                MOV     A,A             ;RETRY LINK
2675  484C  7E             7.0        MOV     A,M             ;GET EXP. DATA
2676  484D                              ROUT    EDATA           ;STORE EXP. DATA
 (1)  484D  D3  95        10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  484F  7F             4.0                MOV     A,A             ;RETRY LINK
2677  4850                              ERRB    ALP01,ALP02,0
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4850  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4853  0F                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4854  00                                .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  4855  CD  15  28     18.0        ALP02:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4858  DA  38  48     10.0                JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
2678                          ;>DATA WRITTEN TO AMTIE NOT = DATA READ
2679  485B  78             4.0        MOV     A,B             ;GET EXP. PTY.
2680  485C  E6  01          7.0        ANI     $01             ;REMOVE UNWANTED BITS
2681  485E  47             4.0        MOV     B,A             ;RESTORE EXP. PTY.
2682  485F  DB  48        10.0        IN      PSTAT           ;GET ACTUAL PTY.
2683  4861  E6  01          7.0        ANI     $01             ;REMOVE UNWANTED BITS
2684  4863  B8             4.0        CMP     B               ;EQUAL?
2685  4864  CA  73  48     10.0        JZ      A1LP02          ;YES
2686  4867                              ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  4867  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4869  7F             4.0                MOV     A,A             ;RETRY LINK
2687  486A  78             4.0        MOV     A,B             ;GET EXP. PTY.
2688  486B                              ROUT    EDATA           ;STORE EXP. PTY.
 (1)  486B  D3  95        10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  486D  7F             4.0                MOV     A,A             ;RETRY LINK
2689  486E                              ERRB    ALP01,A1LP02
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  486E  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0010                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4871  10                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4872  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4873  CD  15  28     18.0        A1LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4876  DA  38  48     10.0                JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
2690                          ;>VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
2691  4879  3E  00          7.0        MVI     A,0             ;CLEAR THE LOOP WRITE TO READ BIT
2692  487B  D3  48        10.0        OUT     PDIAG           ;CLEAR LOOP WRT TO READ BIT
2693
2694  487D  7E             7.0        MOV     A,M             ;GET DATA
2695  487E  3C             4.0        INR     A               ;UPDATE THE DATA BYTE
2696  487F  77             7.0        MOV     M,A             ;SAVE UPDATED DATA
2697  4880  C2  38  48     10.0        JNZ     ALP01           ;CONTINUE UNTIL DONE
2698  4883  C9            10.0        RET
2699
```

```
                                .SBTTL  SUBROUTINE CONTR3 - WMC PARITY GENERATOR
 4884                           SSUB
                                ;*********************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*SUBROUTINE CONTROL 3 - WMC PARITY ERROR GENERATOR
 4884                           SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE CHECKS THE WCS PARITY ERROR INTERRUPT LOGIC FOR THE
                                ;*SPECIFIC TU PORT.
 4884                           SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*     BGNDO
                                ;*     :    CLEAR THE WCS PARITY ERROR INTERRUPT COUNTER
                                ;*     :    ISSUE THE CONTENTS OF COMMAND DATA TO TU PORT COMMAND BYTE
                                ;*     :    INPUT TU PORT STATUS BYTE
                                ;*     :    CALCULATE PARITY ON CONTENTS OF COMMAND DATA
                                ;*     :       IF CALCULATED PARITY IS ODD
                                ;*     :       :    THEN-CONTINUE
                                ;*     :       :    ELSE-IF A PARITY ERROR INTERRUPT
                                ;*     :       :         :    THEN-CONTINUE
                                ;*     :       :         :    ELSE-ERROR
                                ;*     :       :        ENDIF
                                ;*     :       ENDIF
                                ;*     :       IF CALCULATED PARITY WAS ODD
                                ;*     :       :    THEN-IF A PARITY ERROR INTERRUPT
                                ;*     :       :         :    THEN-ERROR
                                ;*     :       :         :    ELSE-CONTINUE
                                ;*     :       :        ENDIF
                                ;*     :       :    ELSE-CONTINUE
                                ;*     :       ENDIF
                                ;*     :    INCREMENT THE TU PORT COMMAND DATA
                                ;*     :    DO UNTIL THE TU PORT COMMAND DATA=ZERO
                                ;*     ENDDO
                                ;*ENDSUB
 4884                           S
                                ;*********************************************************************
 4884  47              4.0     CONTR3: MOV     B,A             ;SAVE THE SELECTED PORT #
 4885  DB  E0          10.0             IN      INTSTA          ;GET THE MASSBUS SELECT DATA
 4887  E6  80           7.0             ANI     BIT7            ;SAVE ONLY SELECT BIT
 4889  B0               4.0             ORA     B               ;ADD IN THE SELECTED PORT #
 488A  D3  E0          10.0             OUT     MBSEL           ;SELECT THE PORT FOR TESTING
 488C  AF               4.0             XRA     A               ;CLEAR INTERRUPT FOR PORT SELECTED
 488D  D3  4C          10.0             OUT     PENAB           ;ENABLE PORT
 488F  3E  60           7.0             MVI     A,P.LCS!P.LWR   ;LOAD LOOP CMD. TO STATUS
                                                                ;       LOOP WRITE TO READ
 4891  D3  48          10.0             OUT     PDIAG           ;SET THE ABOVE MODE TO PORT
 4893  21  4A  49      10.0             LXI     H,DATA          ;LOAD THE ADDRESS OF DATA
 4896  AF               4.0             XRA     A               ;CLEAR THE DATA BYTE
```

```
2745  4897  77                    7.0          MOV     M,A             ;STORE THE DATA
2746  4898  AF                    4.0  TUPTY0: XRA     A               ;CLEAR THE ACC.
2747  4899  32  20  4F            13.0         STA     STPCT           ;CLEAR THE WCS PTY. ERROR COUNT
2748  489C  7E                    7.0          MOV     A,M             ;GET THE DATA
2749  489D                                     ROUT    ADATA           ;STORE IN CAS
 (1)  489D  D3  94                10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  489F  7F                    4.0                  MOV     A,A             ;RETRY LINK
2750  48A0  D3  40                10.0         OUT     TCMD            ;STORE IN THE CMD. REGISTER
2751  48A2  00                    4.0          NOP
2752  48A3  00                    4.0          NOP
2753  48A4  DB  40                10.0         IN      TSTS            ;ACCESS THE TAPE UNIT STATUS BYTE
2754  48A6  7E                    7.0          MOV     A,M             ;GET THE DATA BYTE ISSUED
2755  48A7  A7                    4.0          ANA     A               ;SET THE CONDITION CODE
2756  48A8  E2  DA  48            10.0         JPO     TUPAR0          ;CONTINUE-ODD NUMBER OF ONES
2757  48AB  DB  48                10.0         IN      PSTAT           ;GET THE PORT STATUS WORD
2758  48AD  E6  80                 7.0         ANI     BIT7            ;SAVE STATUS PARITY ERROR BIT
2759  48AF  C2  B7  48            10.0         JNZ     TUCN10          ;JUMP IF OK
2760  48B2                                     ERRA    TUPTY0,TUCN10
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  48B2  CD  0F  28            18.0                 CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0011                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48B5  11                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48B6  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48B7  CD  15  28            18.0       TUCN10::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48BA  DA  98  48            10.0                 JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
2761                                     ;>STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
2762  48BD  3E  80                 7.0          MVI     A,P.INTEN       ;ENABLE FOR INTERRUPT (ALSO CLEAR ERROR BIT)
2763  48BF  D3  4C                10.0         OUT     PENAB
2764  48C1  DB  40                10.0         IN      TSTS            ;CAUSE BIT TO SET AGAIN
2765  48C3  00                    4.0          NOP                     ;DELAY FOR INTERRUPT TO HAPPEN
2766  48C4  00                    4.0          NOP
2767  48C5  3A  20  4F            13.0         LDA     STPCT           ;ELSE-EVEN NUMBER OF ONES
2768  48C8  A7                    4.0          ANA     A               ;THERE SHOULD HAVE BEEN
2769  48C9  C2  D1  48            10.0         JNZ     TUCN11          ;AN INTER.
2770  48CC                                     ERRA    TUPTY0,TUCN11
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  48CC  CD  0F  28            18.0                 CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0012                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48CF  12                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48D0  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48D1  CD  15  28            18.0       TUCN11::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48D4  DA  98  48            10.0                 JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
2771                                     ;>NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
2772  48D7  C3  06  49            10.0         JMP     TUPR10
2773  48DA  DB  48                10.0 TUPAR0: IN      PSTAT           ;GET PORT STATUS WORD
2774  48DC  E6  80                 7.0          ANI     BIT7
2775  48DE  CA  E6  48            10.0         JZ      TUCN20          ;JUMP IF OK
```

```
2777  48E1                                                    ERRA    TUPTY0,TUCN20
 (1)                                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  48E1  CD  0F  28        18.0                                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0013                                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48E4  13                                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48E5  00                                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48E6  CD  15  28        18.0                          TUCN20::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48E9  DA  98  48        10.0                                    JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
2778                                                    ;>STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
2779  48EC  3E  80                7.0                       MVI     A,P.INTEN       ;RE-ENABLE
2780  48EE  D3  4C               10.0                       OUT     PENAB
2781  48F0  DB  40               10.0                       IN      TSTS            ;CAUSE BIT TO SET (IF IT DOES, ITS AN ERROR)
2782  48F2  00                    4.0                       NOP                     ;SMALL DELAY
2783  48F3  00                    4.0                       NOP
2784  48F4  3A  20  4F           13.0                       LDA     STPCT           ;GET THE INTER. COUNT
2785  48F7  A7                    4.0                       ANA     A               ;SET CONDITION BITS
2786  48F8  CA  00  49           10.0                       JZ      TUCN21          ;CONTINUE IF NO INTER.
2787  48FB                                                    ERRA    TUPTY0,TUCN21
 (1)                                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  48FB  CD  0F  28        18.0                                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0014                                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48FE  14                                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48FF  00                                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4900  CD  15  28        18.0                          TUCN21::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4903  DA  98  48        10.0                                    JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
2788                                                    ;>INTER. WHEN NONE EXP.
2789  4906  7E                  7.0                 TUPR10: MOV     A,M             ;GET THE DATA BYTE
2790  4907  3C                  4.0                         INR     A               ;UPDATE THE DATA BYTE
2791  4908  77                  7.0                         MOV     M,A             ;STORE THE UPDATED DATA
2792  4909  AF                  4.0                         XRA     A               ;CLEAR INTERRUPT ENABLE
2793  490A  D3  4C               10.0                        OUT     PENAB
2794  490C  7E                  7.0                         MOV     A,M             ;GET DATA BYTE AGAIN
2795  490D  A7                  4.0                         ANA     A               ;SET CONDITIONS
2796  490E  C2  98  48           10.0                        JNZ     TUPTY0          ;CONTINUE UNTIL DONE
2797  4911  C9                  10.0                        RET
```

```
                                        .SBTTL  SUBROUTINE CONTR4 - LOOP TACH TO TACH
                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*SUBROUTINE CONTROL 4 - LOOP TACH TO TACH
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE CHECKS THE ABILITY OF THE TACH BIT FOR THE SPECIFIED
                                        ;*TU PORT TO SET AND RESET.
                                        SP
                                        ;*********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SET THE TACH AND LOOP W/R BITS FOR SPECIFIED TU PORT
                                        ;*     INPUT THE TACH BIT FOR THE TU PORT
                                        ;*          IF TACH BIT SET
                                        ;*          :    THEN-CONTINUE
                                        ;*          :    ELSE-ERROR
                                        ;*          ENDIF
                                        ;*     CLEAR THE TACH BIT AND SET THE LOOP W/R BITS FOR SPECIFIED TU PORT
                                        ;*     INPUT THE TACH BIT FOR THE TU PORT
                                        ;*          IF TACH BIT RESET
                                        ;*          :    THEN-CONTINUE
                                        ;*          :    ELSE-ERROR
                                        ;*          ENDIF
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************
4912  47              4.0  CONTR4: MOV   B,A                 ;SAVE THE SELECTED PORT #
4913  DB  E0         10.0         IN    INTSTA              ;GET MB SELECT DATA
4915  E6  80          7.0         ANI   BIT7
4917  B0              4.0         ORA   B                   ;ADD IN THE PORT #
4918  D3  E0         10.0         OUT   MBSEL               ;SELECT THE PORT

491A  3E  28          7.0  TACH00: MVI   A,P.TACH!P.LWR    ;LOAD THE TACH BIT AND SET LOOP WRITE TO READ
491C  D3  48         10.0         OUT   PDIAG              ;SET THE TACH BIT
491E  00              4.0         NOP                      ;
491F  DB  48         10.0         IN    PSTAT              ;GET THE PORT STATUS
4921  E6  08          7.0         ANI   P.TACH             ;CHECK THE TACH BIT
4923  C2  28  49     10.0         JNZ   TACH01             ;CONTINUE IF SET
4926                              ERR   TACH00,TACH01
                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
4926  CD  09  28     18.0                 CALL  ERLP               ;PROCESS ERROR - DO 2.3
      0015                          MSGN  =     MSGN+1             ;UPDATE MESSAGE NUMBER FOR THIS
4929  15                            .BYTE MSGN               ;MESSAGE NUMBER ID
492A  00                            .BYTE
492B  CD  15  28     18.0           TACH01::    CALL  CKLOP              ;CHECK LOOP FUNCTION - DO 2.3
492E  DA  1A  49     10.0                 JC    TACH00             ;LOOP ADDRESS IF LOOP SPECIFIED
                                  ;>''TACH'' DID NOT SET
4931  3E  20          7.0  TACH02: MVI   A,P.LWR          ;CLEAR THE TACH BIT
```

```
2836  4933  D3  48        10.0        OUT     PDIAG           ;CLEAR THE TACH BIT
2837  4935  00              4.0        NOP
2838  4936  DB  48        10.0        IN      PSTAT           ;GET THE PORT STATUS
2839  4938  E6  08         7.0        ANI     P.TACH          ;CHECK FOR TACH
2840  493A  CA  42  49     10.0        JZ      TACH03          ;CONTINUE IF ZERO
2841  493D                           ERR     TACH02,TACH03
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  493D  CD  09  28     18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0016                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4940  16                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4941  00                     .BYTE
 (1)  4942  CD  15  28     18.0    TACH03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4945  DA  31  49     10.0                JC      TACH02          ;LOOP ADDRESS IF LOOP SPECIFIED
2842                           ;>''TACH'' DID NOT RESET
2843  4948  C9                10.0        RET
2844
2845
2846
2847                           .SBTTL  PROGRAM VARIABLES
2848
2849  4949  00                UNITMP: .BYTE   0               ;UNIT MAP
2850  494A  00                DATA:   .BYTE   0               ;DATA PATTERN
2851        0000                      .END
```

```
A      =%0007    ADATA  = 0094    ALP01  4838      ALP02  4855  G
AMTIEP= 0001     AMTIE7= 0002     ARAIDF= 0098     ASAVE  4F9B
ATTCD   4F97     AXNUM   4F91     A1LP02 4873  G    B      =%0000
BADST  = 0090    BIT0   = 0001    BIT1   = 0002     BIT15 = 8000
BIT2   = 0004    BIT3   = 0008    BIT4   = 0010     BIT5  = 0020
BIT6   = 0040    BIT7   = 0080    BIT8   = 0100     BIT9  = 0200
BRKHBC= 4F0A     BRKRAM= 4F10     BRKSTR= 4E60     BRKXCT= 4F00
BSAVE   4F9C     BYTCNT= 00D4     BYTEH   4F24     BYTEL  4F23
C      =%0001    CASCT   4F21     CASCTL= 00A0     CASSTA= 00A0
CATTH  = 0089    CATTL  = 0088    CBUSST= 00A1     CBYTH  = 0088
CBYTL  = 008A    CDG1H  = 0087    CDG1L  = 0086    CDG2H  = 0093
CDG2L  = 0092    CDG3H  = 0095    CDG3L  = 0094    CDVTH  = 008D
CDVTL  = 008C    CHPTIE= 0028     CHOTIE= 0020     CH1TIE= 0021
CH2TIE= 0022     CH3TIE= 0023     CH4TIE= 0024     CH5TIE= 0025
CH6TIE= 0026     CH7TIE= 0027     CKLOP  = 2815    CLEAR   4798
CLKCTL= 00F0     CLOCK   4F26     CLRLP   479C     CMC0H  = 0099
CMC0L  = 0098    CMC1H  = 009B    CMC1L  = 009A    CMC2H  = 009D
CMC2L  = 009C    CMC3H  = 009F    CMC3L  = 009E    CMINH  = 0097
CMINL  = 0096    CNTCTL= 00D7     CONTR1  47C3     CONTR2  4827
CONTR3  4884     CONTR4  4912     CRCWRD= 0018     CSAVE   4F9D
CSRLH  = 0091    CSRLL  = 0090    CTCH   = 0085    CTCL   = 0084
CTSTH  = 008F    CTSTL  = 008E    CXCTH  = 0081    CXCTL  = 0080
CXINH  = 0083    CXINL  = 0082    C.     = 0001    C.AVAI= 0080
C.DP   = 0008    C.DSE  = 0010    C.DTU  = 0003    C.DVA  = 0008
C.FAIL= 00FC     C.FMT  = 0070    C.FNCT= 003E     C.GO   = 0001
C.INTC= 00FE     C.MAIN= 0020     C.NSA  = 0080    C.RCT  = 00FC
C.SER  = 0080    C.SHR  = 0040    C.SKPC= 000F     C.TAPE= 0040
C.WCS  = 0002    D      =%0002    DATA    494A     DATACT= 00D0
DBUS    4F28     DBUSCT= 00C0     DBUSST= 00C0     DDRA   = 00D8
DDRAIN= 0010     DDRB   = 00D9    DDRBIN= 0002     DDRC   = 00DA
DDRCIN= 0001     DDRCO  = 0088    DDRCTL= 00DB     DIAFLG  4F22
DIAGPG= 4300     DIAGRM= 4F90     DIARD  = 000B    DONE1  = 0045
DONINT= 0010     DPEN00  4524     DPEN01  453F  G  DPEN02  4545
DPEN03  4560  G  DPEN10  4592     DPEN11  45AD  G  DPEN12  45B3
DPEN13  45CE  G  DPEN20  4615     DPEN21  4630  G  DPEN22  4636
DPEN23  4651  G  DPEN30  4698     DPEN31  46B3  G  DPEN32  46B9
DPEN33  46D4  G  DSAVE   4F9E     DSE    = 0006    DUMMY   431E  G
D.ATH0= 0C01     D.ATH1= 0002     D.EOTD= 0010     D.LAGC= 0020
D.NOTW= 0040     D.NTHR= 0004     D.TACH= 0008     D.WR4  = 0080
E      =%0003    ECCBAD= 0042     ECCCOR= 0019     ECCOK  = 0041
ECCSTA= 001A     ECCTST= 000E     EDATA  = 0095    EOTCLR= 0003
ERFLG   4F93     ERLP   = 2809    ERLPA  = 280F    ERLPB  = 2812
ERLPE  = 280C    ERNUM   4F90     ERRCNT= 00D6     ESAVE   4F9F
EXIT    4792     E.ACRC= 0010     E.AMT  = 0020    E.CDP  = 0080
E.CRC  = 0080    E.PNTR= 0008     E.RPE  = 0040    E.STEC= 0001
E.TTEC= 0002     E.UNC  = 0004    FIFORD= 006A     FORMAT  4F25
FWDTST= 0061     GCRID  = 0089    GCRSET= 0002     GOODTM= 0092
H      =%0004    HLSAVE  4FA0     IE     = 0008    INTSTA= 00E0
ITERA   4F9A     I.PWR  = 0020    I.RMPE= 0040     I5.5   = 0010
I6.5   = 0020    I7.5   = 0040    KCALL  = 005F    KCLR   = 007B
KDEP   = 003F    KENAB  = 0078    KEXAM  = 003E    KEYBRD= 00C8
KEY1   = 0078    KEY10  = 006D    KEY11  = 006E    KEY12  = 006F
KEY13  = 005C    KEY14  = 005D    KEY15  = 005E    KEY16  = 005F
KEY17  = 003C    KEY18  = 003D    KEY19  = 003E    KEY2   = 0079
```

```
KEY20 = 003F            KEY3  = 007A            KEY4  = 007B            KEY5  = 0074
KEY6  = 0075            KEY7  = 0076            KEY8  = 0077            KEY9  = 006C
KINTA = 006F            KLDAD = 003D            KN0   = 003C            KN1   = 005C
KN2   = 005D            KN3   = 005E            KN4   = 006C            KN5   = 006D
KN6   = 006E            KN7   = 0074            KN8   = 0075            KN9   = 0076
KU2   = 0079            KU3   = 007A            KU8   = 0077            L     =%0005
LBLANK= 000F            LCE   = 0008            LCH   = 000C            LCL   = 000D
LCP   = 000E            LC0   = 0000            LC1   = 0001            LC2   = 0002
LC3   = 0003            LC4   = 0004            LC5   = 0005            LC6   = 0006
LC7   = 0007            LC8   = 0008            LC9   = 0009            LDLEDA= 00CA
LDLEDB= 00CB            LDLEDC= 00CC            LDLEDD= 00CD            LDLEDE= 00CE
LDLEDF= 00CF            LKDIAG= 2800            LKKBD = 004C            LKKEY = 0049
LKLWMG= 0058            LKLWMP= 0055            LKLWPG= 0052            LKLWPP= 004F
LKMOD7= 0046            LKOPR = 0046            LPFLG   4F94            LPNUM   4F92
LP01    47D4            LP02    47EB  G         LP03    4805  G         LP04    481A  G
M     =%0006            MBSEL = 00E0            MB.A  = 0008            MB.B  = 0004
MEMTOP= 4FFF            MINUS = 000A            MM    = 8000            MSE   = 0008
MSGN  = 0016            MTACLR= 0000            MT.ARA= 0020            MT.CPE= 0080
MT.DSE= 0001            MT.FWD= 0040            MT.INH= 0008            MT.LWR= 0004
MT.MOT= 0002            MT.NWT= 0080            MT.PEC= 0040            MT.PSB= 0004
MT.PS0= 0001            MT.PS1= 0002            MT.REV= 0020            MT.WRT= 0010
MT.Z  = 0008            M.ATA = 0080            M.CAPE= 0020            M.CONT= 0080
M.DEM = 0020            M.EBL = 0010            M.EXC = 0008            M.FAIL= 0008
M.ILR = 0010            M.INIT= 0010            M.OCC = 0020            M.ONLI= 0001
M.PE  = 0040            M.PORT= 0080            M.RDEN= 0002            M.RDPE= 0008
M.RUN = 0004            M.SCLK= 0001            M.TRA = 0040            M.UNIT= 0007
M.WCLK= 0040            M.WCLN= 0080            M.WREN= 0080            M5.5  = 0001
M6.5  = 0002            M7.5  = 0004            NOTCAP= 0088            OKAY  = 00FF
OPRRAM= 4300            OPSTRT= 0058            OPVER = 0040            PADCNT= 00D5
PADCRC= 0080            PDIAG = 0048            PEID  = 008A            PENAB = 004C
PESET = 0001            PL    = 0081            PRDD  = 004C            PRENF = 009C
PS    = 0082            PSTAT = 0048            PSW   =%0009            P.AMTP= 0001
P.BCTC= 0040            P.CMDP= 0020            P.INTE= 0080            P.LCS = 0040
P.LWR = 0020            P.RDP = 0002            P.RPEN= 0004            P.RPST= 0002
P.RPOE= 0020            P.RP1E= 0010            P.RP2E= 0020            P.RP3E= 0010
P.SING= 0080            P.STAT= 0002            P.STPE= 0080            P.TACH= 0008
P.TUPR= 0010            P.WCSP= 0004            P.WDS = 0040            P.WFLP= 0001
P.WPEN= 0010            P.WPOE= 0008            P.WP1E= 0004            P.WP2E= 0008
P.WP3E= 0004            P.5VOK= 0002            QUE   = 281B            QUEM  = 281E
RAMT  = 0010            RARA  = 0006            RARAI = 0004            RCHBD0= 0048
RCHBD1= 0047            RCHOK = 0046            RCHTST= 000C            RCLRT = 000D
RCMD  = 0008            RCMLP = 0003            RCONT = 0080            RDATA = 0017
RDCLK = 0010            RDON  = 0011            READG = 0007            REND  = 0014
REQST = 2806            RESCHR= 00D1            REVTST= 0064            REWIND= 0004
RFIFOL= 0008            RGCLK = 0002            RGCRI = 0003            RIBG  = 0001
RILL  = 0012            RINST = 000C            RMCTST= 0008            RMK2  = 0013
RNOP  = 0000            RPATH = 0001            RPBAD = 0044            RPCHI = 0001
RPCLK = 0003            RPCTL = 0009            RPEI  = 0002            RPFAIL= 0000
RPF1  = 009D            RPF2  = 009E            RPOK  = 0043            RPOSTN= 0016
RPSTA = 0015            RRCMT = 000A            RSTAT = 0002            RTIEB = 000A
RTIER = 0030            RTM   = 0005            RUNKI = 0009            RUPTST= 005E
RWDUNL= 0005            R.AMT = 0001            R.BOP = 0008            R.DATA= 0040
R.DON = 0002            R.DRDY= 0010            R.END = 0010            R.ILL = 0004
R.JVOK= 0004            R.MK2 = 0008            R.PLOD= 0008            R.PLOO= 0010
```

```
R.PL01= 0020          R.POST= 0020          R.STNM= 0002          R.STOP= 0004
R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040          R.VOK = 0080
R00H  = 0081          R00L  = 0080          R01H  = 0083          R01L  = 0082
R02H  = 0085          R02L  = 0084          R03H  = 0087          R03L  = 0086
R04H  = 0089          R04L  = 0088          R05H  = 008B          R05L  = 008A
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          TACH00  491A          TACH01  492B  G       TACH02  4931
TACH03  4942  G       TADR00= 0080          TADR01= 0081          TADR02= 0082
TADR03= 0083          TADR04= 0084          TADR05= 0085          TADR06= 0086
TADR07= 0087          TADR10= 0088          TADR11= 0089          TADR12= 008A
TADR13= 008B          TAMT  = 0044          TASEL = 0080          TCMD  = 0040
TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020
TC.WRT= 0010          TEMP    4F99          TEST1   4300          TEST10  443F
TEST11  4468          TEST12  4491          TEST13  44BA          TEST14  44E3
TEST15  450C          TEST16  457A          TEST17  45E8          TEST2   434A
TEST20  466B          TEST21  46EE          TEST22  4717          TEST23  4740
TEST24  4769          TEST3   4373          TEST4   439C          TEST5   43C5
TEST6   43ED          TEST7   4416          TMF   = 0099          TMRDY = 0040
TRKENA= 00D2          TSET  = 2803          TSTEND= 2818          TSTS  = 0040
TST01X  4327          TST02X  434F          TST03X  4378          TST04X  43A1
TST05X  43CA          TST06X  43F2          TST07X  441B          TST10X  4444
TST11X  446D          TST12X  4496          TST13X  44BF          TST14X  44E8
TST15X  4511          TST16X  457F          TST17P  460F  G       TST17X  45ED
TST20P  4692  G       TST20X  4670          TST21X  46F3          TST22X  471C
TST23X  4745          TST24X  476E          TUCN10  48B7  G       TUCN11  48D1  G
TUCN20  48E6  G       TUCN21  4900  G       TUPAR0  48DA          TUPR10  4906
TUPTY0  4898          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDY0= 0040          T.RWD = 0010          T.SCLK= 0002          UIBG  = 00A1
UNITMP  4949          UNIT00  480E          UNIT0X  4820          VALFC   4F98
VALTB   4F95          VELTST= 005B          WDR.P = 0010          WMCCTL= 00D3
WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3
W.ACRC= 0004          W.CRC = 0008          W.DIAG= 0002          W.DONN= 0040
W.ECC = 0010          W.ENAB= 0080          W.ERR = 0020          W.FMT = 0070
W.GCR = 0010          W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002
W.REV = 0004          W.ROME= 0010          W.RST = 0001          W.SKIP= 000F
W.WRIT= 0008          W.XFER= 0020          X     =%000A          X.DONN= 0080
X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001
Y     =%0008          .     = 494B


ERRORS DETECTED:  0

*TUP1.A78/PTP,TUP1=NLIST,PARAM,MACRO,LIST,TUP1
```

RUN-TIME: 5 8 0 SECONDS
CORE USED: 10K

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:02
PARAM.M80 TABLE OF CONTENTS SEQ 0287

```
                                .TITLE  MTA1 - MAG TAPE ADAPTER TEST PART #1
                                ;ID     MTA1-MAG TAPE ADAPTER CONTROLLER PART #1

                                .SBTTL  TEST 01 - LOOP COMMAND TO STATUS - MTA #0
4300                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS TEST-MTA #0
4300                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
                                ;*MTA #0.
4300                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #0 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 0
                                ;*     IF TU PRESENT FOR TU PORT 0 = 0
                                ;*     :   THEN-ERROR 02
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #0
                                ;*     INPUT THE STATUS BYTE A FOR MTA #0
                                ;*     IF FWD, REV, WRT, WRT INH AND LWR BITS SET
                                ;*     :   THEN-CONTINUE
                                ;*     :   ELSE-ERROR 03
                                ;*     ENDIF
                                ;*     RESET THE COMMAND BITS FOR MTA #0
                                ;*     INPUT THE STATUS BYTE A FOR MTA #0
                                ;*     MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
                                ;*     IF ALL BITS OF RESULT RESET
                                ;*     :   THEN-CONTINUE
                                ;*     :   ELSE-ERROR 04
                                ;*     ENDIF
                                ;*ENDTST
4300                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 01
                                ;*MTA1 MICRO ERROR 01
                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
                                ;*M8954, M8955'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - MICRO MODULE ABORTED
```

```
1371                                    ;*
1372                                    ;*MTA1 MICRO TEST 01
1373                                    ;*MTA1 MICRO ERROR 02
1374                                    ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
1375                                    ;*M8954, M8955'S
1376                                    ;*PORT 0 'TU PRES' BIT NOT SET
1377                                    ;*
1378                                    ;*MTA1 MICRO TEST 01
1379                                    ;*MTA1 MICRO ERROR 03
1380                                    ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
1381                                    ;*M8954, M8955'S
1382                                    ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1383                                    ;*ACTUAL = NNNN
1384                                    ;*EXPECTED = NNNN
1385                                    ;*
1386                                    ;*MTA1 MICRO TEST 01
1387                                    ;*MTA1 MICRO ERROR 04
1388                                    ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
1389                                    ;*M8954, M8955'S
1390                                    ;*STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1391  4300                              S
 (1)                                    ; ***************************************************************************
1392
1393  4300                              TEST01: TESTX   @01              ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                 MVI     A,@01               ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                 CALL    TSET                ;SETUP THE TEST
1394                                    ;%MTA1-LOOP COMMAND TO STATUS-MTA #0
1395                                    ;&M8954, M8955'S
1396  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28       18.0                         CALL    REQST
 (1)  4308  00                                            .BYTE   0                   ;DATA PATTERN NUMBER
 (1)  4309  00  00                                        .WORD   0                   ;SYSTEM '0'' COUNT
 (1)  430B  00  00                                        .WORD   0                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                            .BYTE   0                   ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                            .BYTE   @7                  ;REQUEST CODE
1397  430F                                RIN     R12L                ;GET THE UNITS DESIRED
 (1)  430F  DB  94            10.0                IN      R12L                ;READ R12L INTO AC
 (1)  4311  7F                 4.0                 MOV     A,A                 ;RETRY LINK
1398  4312  32  CD  49        13.0          STA     UNITMP              ;STORE IN MEMORY
1399  4315  A7                  4.0         ANA     A                   ;SET THE CONDITION CODE
1400  4316  C2  27  43        10.0          JNZ     TST01X              ;GO TEST THE SPECIFIED UNITS
1401                                                                    ;NO UNITS SPECIFIED
1402  4319                                ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28       18.0                 CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0001                               MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                 .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  431D  00                                 .BYTE
 (1)  431E  CD  15  28       18.0          DUMMY:: CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  9C  49       10.0                  JC      EXIT                ;LOOP ADDRESS IF LOOP SPECIFIED
1403                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1404                                    ;>FATAL ERROR - MICRO MODULE ABORTED
1405  4324  C3  9C  49       10.0          JMP     EXIT                ;EXIT THE TEST
1406
```

```
1407  4327  3A  CD  49      13.0  TST01X: LDA   UNITMP          ;GET THE UNIT MAP
1408  432A  E6  01            7.0          ANI   @01            ;TEST FOR UNIT 0
1409  432C  CA  A5  43       10.0          JZ    TEST02         ;NO-GO CHECK FOR TEST 02
1410                                       ;
1411  432F  CD  A2  49       18.0          CALL  CLEAR          ;CLEAN UP THE PORTS
1412  4332  DB  E0           10.0          IN    INTSTA         ;GET MB SELECT STATUS
1413  4334  E6  80            7.0          ANI   BIT7           ;
1414  4336  D3  E0           10.0          OUT   MBSEL          ;SELECT PORT 0
1415  4338  DB  48           10.0          IN    PSTAT          ;GET TAPE STATUS TO SEF IF A UNIT IS PRESENT
1416  433A  E6  10            7.0          ANI   P.TUPRES       ;SAVE ONLY PRES BIT
1417  433C  C2  44  43        10.0          JNZ   TST01C         ;CONTINUE IF ITS SET
1418  433F                                 ERR   TST01X,TST01C
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  433F  CD  09  28        18.0                CALL  ERLP           ;PROCESS ERROR - DO 2.3
 (1)        0002                          MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4342  02                            .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  4343  00                            .BYTE
 (1)  4344  CD  15  28        18.0  TST01C::      CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4347  DA  27  43        10.0                JC    TST01X         ;LOOP ADDRESS IF LOOP SPECIFIED
1419                                 ;>PORT 0 'TU PRES' BIT NOT SET
1420
1421  434A  3E  80            7.0  DPEN00: MVI   A,$80          ;LOAD MTA #0 REGISTER 0 SELECT CODE
1422  434C  D3  40           10.0          OUT   TCMD
1423  434E  3E  7C            7.0          MVI   A,$7C          ;SET THE MTA #0 COMMAND BITS
1424  4350                                 ROUT  EDATA          ;
 (1)  4350  D3  95           10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  4352  7F                4.0                MOV   A,A            ;RETRY LINK
1425  4353  D3  40           10.0          OUT   TCMD
1426  4355  00                4.0          NOP                  ;
1427  4356  3E  81            7.0          MVI   A,$81          ;LOAD MTA #0 REGISTER 1 SELECT CODE
1428  4358  D3  40           10.0          OUT   TCMD
1429  435A  00                4.0          NOP                  ;WAIT
1430  435B  DB  40           10.0          IN    TSTS           ;INPUT THE MTA STATUS A BYTE
1431  435D  E6  7C            7.0          ANI   $7C            ;REMOVE UNWANTED BITS
1432  435F                                 ROUT  ADATA          ;
 (1)  435F  D3  94           10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  4361  7F                4.0                MOV   A,A            ;RETRY LINK
1433  4362  FE  7C            7.0          CPI   $7C            ;
1434  4364  CA  6C  43        10.0          JZ    DPEN01         ;CONTINUE IF SET
1435  4367                                 ERRB  DPEN00,DPEN01
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4367  CD  12  28        18.0                CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)        0003                          MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  436A  03                            .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  436B  00                            .BYTE                ;PRINT ROUTINE NUMBER
 (1)  436C  CD  15  28        18.0  DPEN01::      CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  436F  DA  4A  43        10.0                JC    DPEN00         ;LOOP ADDRESS IF LOOP SPECIFIED
1436                                 ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1437  4372  3E  80            7.0  DPEN02: MVI   A,$80          ;LOAD MTA #0 REGISTER 0 SELECT CODE
1438  4374  D3  40           10.0          OUT   TCMD
1439  4376  AF                4.0          XRA   A              ;CLEAR THE MTA #0 LOOP WRITE/READ BIT
1440  4377  D3  40           10.0          OUT   TCMD
1441  4379  00                4.0          NOP
1442  437A  3E  81            7.0          MVI   A,$81          ;LOAD MTA #0 REGISTER 1 SELECT CODE
```

```
1443  437C  D3  40          10.0        OUT     TCMD
1444  437E  00               4.0        NOP                     ;WAIT
1445  437F  DB  40          10.0        IN      TSTS            ;INPUT THE MTA STATUS A BYTE
1446  4381  E6  04           7.0        ANI     $04             ;CHECK FOR LWR RESET
1447  4383  CA  8B  43      10.0        JZ      DPEN03          ;CONTINUE IF ZERO
1448  4386                             ERR     DPEN02,DPEN03
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4386  CD  09  28      18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4389  04                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  438A  00                        .BYTE
 (1)  438B  CD  15  28      18.0      DPEN03::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  438E  DA  72  43      10.0                JC      DPEN02          ;LOOP ADDRESS IF LOOP SPECIFIED
1449                        ;>STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1450                                            ;
1451  4391                      ENDTST  TST01X
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4391                  REQ     7           ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4391  CD  06  28      18.0        CALL    REQST
 (2)  4394  00                          .BYTE               ;DATA PATTERN NUMBER
 (2)  4395  00  00                      .WORD               ;SYSTEM '@' COUNT
 (2)  4397  00  00                      .WORD               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4399  00                          .BYTE               ;DATA COMPARE FLAG IF =1
 (2)  439A  07                          .BYTE   7           ;REQUEST CODE
 (1)  439B  3A  9A  4F      13.0        LDA     ITERA       ;GET ITERATION COUNT
 (1)  439E  3D               4.0        DCR     A           ;DOWNCOUNT
 (1)  439F  32  9A  4F      13.0        STA     ITERA       ;SAVE COUNT
 (1)  43A2  F2  27  43      10.0        JP      TST01X      ;DO TEST UNTIL TILL = 0
1452
1453
```

```
                                .SBTTL  TEST 02 - LOOP COMMAND TO STATUS - MTA #1

                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS TEST-MTA #1
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
                                ;*MTA #1.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 1
                                ;*    IF TU PRESENT FOR TU PORT 1 = 0
                                ;*    :   THEN-ERROR 05
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #1
                                ;*    INPUT THE STATUS BYTE A FOR MTA #1
                                ;*    IF FWD, REV, WRT, WRT INH AND LWR BITS SET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 06
                                ;*    ENDIF
                                ;*    RESET THE COMMAND BITS FOR MTA #1
                                ;*    INPUT THE STATUS BYTE A FOR MTA #1
                                ;*    MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
                                ;*    IF ALL BITS OF RESULT RESET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 07
                                ;*    ENDIF
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 02
                                ;*MTA1 MICRO ERROR 05
                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #1
                                ;*M8954, M8955'S
                                ;*PORT 1 'TU PRES' BIT NOT SET
                                ;*
                                ;*MTA1 MICRO TEST 02
                                ;*MTA1 MICRO ERROR 06
```

```
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #1
                                        ;*M8954, M8955'S
                                        ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 02
                                        ;*MTA1 MICRO ERROR 07
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #1
                                        ;*M8954, M8955'S
                                        ;*STATUS BYTE NOT = ZERO WHEN COMMAND = 0
43A5                                    S
 (1)                                    ; ***************************************************************************
                                        TEST02: TESTX   @02             ;INITIALIZE THE TEST
 (1) 43A5  3E  02          7.0                   MVI     A,@02           ;DEFINE THE TEST NUMBER
 (1) 43A7  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP COMMAND TO STATUS-MTA #1
                                        ;&M8954, M8955'S

43AA  3A  CD  49         13.0  TST02X: LDA     UNITMP          ;GET THE UNIT MAP
43AD  E6  02              7.0        ANI     @02             ;TEST FOR UNIT 1
43AF  CA  2A  44         10.0        JZ      TEST03          ;NO-CHECK FOR TEST 03
                                     ;
43B2  CD  A2  49         18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
43B5  DB  E0             10.0        IN      INTSTA          ;GET SELECT STATUS
43B7  E6  80              7.0        ANI     BIT7
43B9  F6  01              7.0        ORI     1
43BB  D3  E0             10.0        OUT     MBSEL           ;SELECT PORT 1

43BD  DB  48             10.0        IN      PSTAT           ;GET TAPE STATUS TO SEE IF A UNIT IS PRESENT
43BF  E6  10              7.0        ANI     P.TUPRES        ;SAVE ONLY PRES BIT
43C1  C2  C9  43         10.0        JNZ     TST02C          ;CONTINUE IF ITS SET
43C4                                 ERR     TST02X,TST02C
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 43C4  CD  09  28     18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 43C7  05                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 43C8  00                                    .BYTE
 (1) 43C9  CD  15  28     18.0           TST02C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 43CC  DA  AA  43     10.0                   JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                     ;>PORT 1 'TU PRES' BIT NOT SET
43CF  3E  80              7.0  DPEN10: MVI     A,$80
43D1  D3  40             10.0        OUT     TCMD
43D3  3E  7C              7.0        MVI     A,$7C           ;SET THE MTA #0 COMMAND BITS
43D5                                 ROUT    EDATA           ;
 (1) 43D5  D3  95         10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1) 43D7  7F              4.0                   MOV     A,A             ;RETRY LINK
43D8  D3  40             10.0        OUT     TCMD
43DA  00                  4.0        NOP
43DB  3E  81              7.0        MVI     A,$81
43DD  D3  40             10.0        OUT     TCMD
43DF  00                  4.0        NOP             ;
                                     ;
```

```
1539  43E0  DB  40          10.0          IN      TSTS            ;GET THE MTA STATUS A
1540  43E2  E6  7C           7.0          ANI     $7C             ;REMOVE UNWANTED BITS
1541  43E4                               ROUT    ADATA           ;
 (1)  43E4  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43E6  7F               4.0                  MOV     A,A             ;RETRY LINK
1542  43E7  FE  7C           7.0          CPI     $7C             ;
1543  43E9  CA  F1  43       10.0          JZ      DPEN11          ;CONTINUE IF SET
1544  43EC                               ERRB    DPEN10,DPEN11
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43EC  CD  12  28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43EF  06                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43F0  00                                     .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43F1  CD  15  28       18.0          DPEN11::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43F4  DA  CF  43       10.0                  JC      DPEN10          ;LOOP ADDRESS IF LOOP SPECIFIED
1545                               ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1546  43F7  3E  80           7.0  DPEN12: MVI     A,$80           ;LOAD MTA #1 REGISTER 0 SELECT CODE
1547  43F9  D3  40          10.0          OUT     TCMD            ;
1548  43FB  AF               4.0          XRA     A               ;CLEAR THE MTA #1 LOOP WRITE/READ BIT
1549  43FC  D3  40          10.0          OUT     TCMD
1550  43FE  00               4.0          NOP
1551  43FF  3E  81           7.0          MVI     A,$81           ;LOAD MTA #1 REGISTER 1 SELECT CODE
1552  4401  D3  40          10.0          OUT     TCMD
1553  4403  00               4.0          NOP                     ;WAIT
1554  4404  DB  40          10.0          IN      TSTS            ;INPUT THE MTA STATUS A BYTE
1555  4406  E6  04           7.0          ANI     $04             ;CHECK FOR LWR RESET
1556  4408  CA  10  44       10.0          JZ      DPEN13          ;CONTINUE IF ZERO
1557  4408                               ERR     DPEN12,DPEN13
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4408  CD  09  28       18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  440E  07                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  440F  00                                     .BYTE
 (1)  4410  CD  15  28       18.0          DPEN13::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4413  DA  F7  43       10.0                  JC      DPEN12          ;LOOP ADDRESS IF LOOP SPECIFIED
1558                               ;>STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1559
1560  4416                                 ENDTST  TST02X
 (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4416                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4416  CD  06  28       18.0                  CALL    REQST
 (2)  4419  00                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  441A  00  00                                 .WORD                   ;SYSTEM '#' COUNT
 (2)  441C  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  441E  00                                     .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  441F  07                                     .BYTE   7               ;REQUEST CODE
 (1)  4420  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4423  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4424  32  9A  4F       13.0          STA     ITERA           ;SAVE COUNT
 (1)  4427  F2  AA  43       10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1561
1562
```

```
                                .SBTTL  TEST 03 - LOOP COMMAND TO STATUS - MTA #2

                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS TEST-MTA #2
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
                                ;*MTA #2.
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #2 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 2
                                ;*     IF TU PRESENT FOR TU PORT 2 = 0
                                ;*     :   THEN-ERROR 10
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #2
                                ;*     INPUT THE STATUS BYTE A FOR MTA #2
                                ;*     IF FWD, REV, WRT, WRT INH AND LWR BITS SET
                                ;*     :   THEN-CONTINUE
                                ;*     :   ELSE-ERROR 11
                                ;*     ENDIF
                                ;*     RESET THE COMMAND BITS FOR MTA #2
                                ;*     INPUT THE STATUS BYTE A FOR MTA #2
                                ;*     MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
                                ;*     IF ALL BITS OF RESULT RESET
                                ;*     :   THEN-CONTINUE
                                ;*     :   ELSE-ERROR 12
                                ;*     ENDIF
                                ;*ENDTST
                                SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 03
                                ;*MTA1 MICRO ERROR 10
                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #2
                                ;*M8954, M8955'S
                                ;*PORT 2 'TU PRES' BIT NOT SET
                                ;*
                                ;*MTA1 MICRO TEST 03
                                ;*MTA1 MICRO ERROR 11
```

```
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #2
                                        ;*M8954, M8955'S
                                        ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 03
                                        ;*MTA1 MICRO ERROR 12
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #2
                                        ;*M8954, M8955'S
                                        ;*STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1617  442A                              S
 (1)                                    ; ***************************************************************************
1618
1619  442A                              TEST03: TESTX   @03               ;INITIALIZE THE TEST
 (1)  442A  3E  03            7.0                 MVI     A,@03          ;DEFINE THE TEST NUMBER
 (1)  442C  CD  03  28       18.0                 CALL    TSET           ;SETUP THE TEST
1620                                    ;%MTA1-LOOP COMMAND TO STATUS-MTA #2
1621                                    ;&M8954, M8955'S
1622
1623  442F  3A  CD  49        13.0  TST03X: LDA     UNITMP                 ;GET THE UNIT MAP
1624  4432  E6  04             7.0         ANI     @04                    ;TEST FOR UNIT 2
1625  4434  CA  AF  44        10.0         JZ      TEST04                 ;NO-GO CHECK FOR TEST 04
1626                                                                       ;
1627  4437  CD  A2  49        18.0         CALL    CLEAR                  ;CLEAN UP THE PORTS
1628  443A  DB  E0             10.0         IN      INTSTA                 ;GET SELECT STATUS
1629  443C  E6  80              7.0         ANI     BIT7
1630  443E  F6  02              7.0         ORI     2
1631  4440  D3  E0             10.0         OUT     MBSEL                  ;SELECT PORT 2
1632
1633  4442  DB  48             10.0         IN      PSTAT                  ;GET TAPE STATUS TO SEE IF A UNIT IS PRESENT
1634  4444  E6  10              7.0         ANI     P.TUPRES               ;SAVE ONLY PRES BIT
1635  4446  C2  4E  44         10.0         JNZ     TST03C                 ;CONTINUE IF ITS SET
1636  4449                               ERR     TST03X,TST03C
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4449  CD  09  28       18.0                 CALL    ERLP           ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  444C  08                                .BYTE   MSGN           ;MESSAGE NUMBER ID
 (1)  444D  00                                .BYTE
 (1)  444E  CD  15  28       18.0  TST03C::       CALL    CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4451  DA  2F  44       10.0                 JC      TST03X         ;LOOP ADDRESS IF LOOP SPECIFIED
1637                                    ;>PORT 2 'TU PRES' BIT NOT SET
1638
1639  4454  3E  80              7.0  DPEN20: MVI     A,$80
1640  4456  D3  40             10.0         OUT     TCMD
1641  4458  3E  7C              7.0         MVI     A,$7C                  ;SET THE MTA #2 COMMAND BITS
1642  445A                               ROUT    EDATA                  ;
 (1)  445A  D3  95             10.0                 OUT     EDATA          ;WRITE AC INTO EDATA
 (1)  445C  7F                  4.0                 MOV     A,A            ;RETRY LINK
1643  445D  D3  40             10.0         OUT     TCMD
1644  445F  00                  4.0         NOP
1645  4460  3E  81              7.0         MVI     A,$81
1646  4462  D3  40             10.0         OUT     TCMD
1647  4464  00                  4.0         NOP                            ;
```

```
1648                                                    ;
1649  4465  DB  40          10.0      IN    TSTS        ;GET THE MTA STATUS A
1650  4467  E6  7C           7.0      ANI   $7C         ;REMOVE UNWANTED BITS
1651  4469                           ROUT  ADATA       ;
 (1)  4469  D3  94          10.0            OUT   ADATA       ;WRITE AC INTO ADATA
 (1)  4468  7F               4.0            MOV   A,A         ;RETRY LINK
1652  446C  FE  7C           7.0      CPI   $7C         ;
1653  446E  CA  76  44       10.0      JZ    DPEN21      ;CONTINUE IF SET
1654  4471                           ERRB  DPEN20,DPEN21
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4471  CD  12  28      18.0                  CALL  ERLPB       ;PROCESS ERROR - DO 2.3
 (1)       0009                               MSGN  =     MSGN+1   ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4474  09                                 .BYTE MSGN        ;MESSAGE NUMBER ID
 (1)  4475  00                                 .BYTE             ;PRINT ROUTINE NUMBER
 (1)  4476  CD  15  28      18.0          DPEN21::  CALL  CKLOP       ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4479  DA  54  44      10.0                  JC    DPEN20      ;LOOP ADDRESS IF LOOP SPECIFIED
1655                           ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1656  447C  3E  80           7.0 DPEN22: MVI   A,$80       ;LOAD MTA #2 REGISTER 0 SELECT CODE
1657  447E  D3  40          10.0      OUT   TCMD        ;
1658  4480  AF               4.0      XRA   A           ;CLEAR THE MTA #2 LOOP WRITE/READ BIT
1659  4481  D3  40          10.0      OUT   TCMD
1660  4483  00               4.0      NOP
1661  4484  3E  81           7.0      MVI   A,$81       ;LOAD MTA #2 REGISTER 1 SELECT CODE
1662  4486  D3  40          10.0      OUT   TCMD
1663  4488  00               4.0      NOP               ;WAIT
1664  4489  DB  40          10.0      IN    TSTS        ;INPUT THE MTA STATUS A BYTE
1665  448B  E6  04           7.0      ANI   $04         ;CHECK FOR LWR RESET
1666  448D  CA  95  44       10.0      JZ    DPEN23      ;CONTINUE IF ZERO
1667  4490                           ERR   DPEN22,DPEN23
 (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4490  CD  09  28      18.0                  CALL  ERLP        ;PROCESS ERROR - DO 2.3
 (1)       000A                               MSGN  =     MSGN+1   ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4493  0A                                 .BYTE MSGN        ;MESSAGE NUMBER ID
 (1)  4494  00                                 .BYTE
 (1)  4495  CD  15  28      18.0          DPEN23::  CALL  CKLOP       ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4498  DA  7C  44      10.0                  JC    DPEN22      ;LOOP ADDRESS IF LOOP SPECIFIED
1668                           ;>STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1669
1670  449B                           ENDTST  TST03X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  449B                          REQ     7                 ;FAKE CALL TO KEEP TEST ALIVE
 (2)  449B  CD  06  28      18.0                  CALL  REQST
 (2)  449E  00                                 .BYTE             ;DATA PATTERN NUMBER
 (2)  449F  00  00                             .WORD             ;SYSTEM '*' COUNT
 (2)  44A1  00  00                             .WORD             ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44A3  00                                 .BYTE             ;DATA COMPARE FLAG IF =1
 (2)  44A4  07                                 .BYTE 7           ;REQUEST CODE
 (1)  44A5  3A  9A  4F      13.0      LDA   ITERA       ;GET ITERATION COUNT
 (1)  44A8  3D               4.0      DCR   A           ;DOWNCOUNT
 (1)  44A9  32  9A  4F      13.0      STA   ITERA       ;SAVE COUNT
 (1)  44AC  F2  2F  44      10.0      JP    TST03X      ;DO TEST UNTIL TILL = 0
1671
1672
```

```
                                .SBTTL  TEST 04 - LOOP COMMAND TO STATUS - MTA #3

44AF                            ST
                                ;*************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS TEST-MTA #3
44AF                            SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
                                ;*MTA #3.
44AF                            SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 3
                                ;*    IF TU PRESENT FOR TU PORT 3 = 0
                                ;*    :   THEN-ERROR 13
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #3
                                ;*    INPUT THE STATUS BYTE A FOR MTA #3
                                ;*    IF FWD, REV, WRT, WRT INH AND LWR BITS SET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 14
                                ;*    ENDIF
                                ;*    RESET THE COMMAND BITS FOR MTA #3
                                ;*    INPUT THE STATUS BYTE A FOR MTA #3
                                ;*    MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
                                ;*    IF ALL BITS OF RESULT RESET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 15
                                ;*    ENDIF
                                ;*ENDTST
44AF                            SE
                                ;*************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 04
                                ;*MTA1 MICRO ERROR 13
                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #3
                                ;*M8954, M8955'S
                                ;*PORT 3 'TU PRES' BIT NOT SET
                                ;*
                                ;*MTA1 MICRO TEST 04
                                ;*MTA1 MICRO ERROR 14
```

```
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #3
                                        ;*M8954, M8955'S
                                        ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 04
                                        ;*MTA1 MICRO ERROR 15
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #3
                                        ;*M8954, M8955'S
                                        ;*STATUS BYTE NOT = ZERO WHEN COMMAND = 0
44AF                                    S
 (1)                                    ; ***************************************************************************
                                        TEST04: TESTX   @04             ;INITIALIZE THE TEST
 (1) 44AF  3E  04          7.0                   MVI     A,@04           ;DEFINE THE TEST NUMBER
 (1) 44B1  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP COMMAND TO STATUS-MTA #3
                                        ;&M8954, M8955'S

44B4  3A  CD  49         13.0  TST04X: LDA     UNITMP          ;GET THE UNIT MAP
44B7  E6  08              7.0         ANI     @010            ;TEST FOR UNIT 3
44B9  CA  34  45         10.0         JZ      TEST05          ;NO-GO CHECK FOR TEST 05
                                                              ;
44BC  CD  A2  49         18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
44BF  DB  E0             10.0         IN      INTSTA          ;GET SELECT STATUS
44C1  E6  80              7.0         ANI     BIT7
44C3  F6  03              7.0         ORI     3
44C5  D3  E0             10.0         OUT     MBSEL           ;SELECT PORT 3

44C7  DB  48             10.0         IN      PSTAT           ;GET TAPE STATUS TO SEE IF A UNIT IS PRESENT
44C9  E6  10              7.0         ANI     P.TUPRES        ;SAVE ONLY PRES BIT
44CB  C2  D3  44         10.0         JNZ     TST04C          ;CONTINUE IF ITS SET
44CE                                   ERR     TST04X,TST04C
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 44CE  CD  09  28     18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000B                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 44D1  0B                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 44D2  00                                   .BYTE
 (1) 44D3  CD  15  28     18.0          TST04C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 44D6  DA  B4  44     10.0                   JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>PORT 3 'TU PRES' BIT NOT SET

44D9  3E  80              7.0  DPEN30: MVI     A,$80
44DB  D3  40             10.0         OUT     TCMD
44DD  3E  7C              7.0         MVI     A,$7C           ;SET THE MTA #3 COMMAND BITS
44DF                                   ROUT    EDATA           ;
 (1) 44DF  D3  95         10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1) 44E1  7F              4.0                   MOV     A,A             ;RETRY LINK
44E2  D3  40             10.0         OUT     TCMD
44E4  00                  4.0         NOP
44E5  3E  81              7.0         MVI     A,$81
44E7  D3  40             10.0         OUT     TCMD
44E9  00                  4.0         NOP                     ;
```

```
1758                                                    ;
1759  44EA  DB  40          10.0       IN    TSTS       ;GET THE MTA STATUS A
1760  44EC  E6  7C           7.0       ANI   $7C        ;REMOVE UNWANTED BITS
1761  44EE                             ROUT  ADATA      ;
 (1)  44EE  D3  94          10.0             OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  44F0  7F               4.0             MOV   A,A            ;RETRY LINK
1762  44F1  FE  7C           7.0       CPI   $7C        ;
1763  44F3  CA  FB  44       10.0       JZ    DPEN31     ;CONTINUE IF SET
1764  44F6                             ERRB  DPEN30,DPEN31
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44F6  CD  12  28       18.0                 CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)      000C                                    MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F9  0C                                    .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  44FA  00                                    .BYTE                ;PRINT ROUTINE NUMBER
 (1)  44FB  CD  15  28       18.0           DPEN31::  CALL  CKLOP      ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FE  DA  D9  44       10.0                 JC    DPEN30         ;LOOP ADDRESS IF LOOP SPECIFIED
1765                          ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1766  4501  3E  80           7.0  DPEN32: MVI   A,$80      ;LOAD MTA #3 REGISTER 0 SELECT CODE
1767  4503  D3  40          10.0       OUT   TCMD       ;
1768  4505  AF               4.0       XRA   A          ;CLEAR THE MTA #3 LOOP WRITE/READ BIT
1769  4506  D3  40          10.0       OUT   TCMD
1770  4508  00               4.0       NOP
1771  4509  3E  81           7.0       MVI   A,$81      ;LOAD MTA #3 REGISTER 1 SELECT CODE
1772  450B  D3  40          10.0       OUT   TCMD
1773  450D  00               4.0       NOP              ;WAIT
1774  450E  DB  40          10.0       IN    TSTS       ;INPUT THE MTA STATUS A BYTE
1775  4510  E6  04           7.0       ANI   $04        ;CHECK FOR LWR RESET
1776  4512  CA  1A  45       10.0       JZ    DPEN33     ;CONTINUE IF ZERO
1777  4515                             ERR   DPEN32,DPEN33
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4515  CD  09  28       18.0                 CALL  ERL?           ;PROCESS ERROR - DO 2.3
 (1)      000D                                    MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4518  0D                                    .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  4519  00                                    .BYTE
 (1)  451A  CD  15  28       18.0           DPEN33::  CALL  CKLOP      ;CHECK LOOP FUNCTION - DO 2.3
 (1)  451D  DA  01  45       10.0                 JC    DPEN32         ;LOOP ADDRESS IF LOOP SPECIFIED
1778                          ;>STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1779
1780  4520                             ENDTST  TST04X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4520                             REQ   7          ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4520  CD  06  28       18.0                 CALL  REQST
 (2)  4523  00                                    .BYTE                ;DATA PATTERN NUMBER
 (2)  4524  00  00                                .WORD                ;SYSTEM '*' COUNT
 (2)  4526  00  00                                .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4528  00                                    .BYTE                ;DATA COMPARE FLAG IF =1
 (2)  4529  07                                    .BYTE  7             ;REQUEST CODE
 (1)  452A  3A  9A  4F       13.0       LDA   ITERA      ;GET ITERATION COUNT
 (1)  452D  3D               4.0       DCR   A          ;DOWNCOUNT
 (1)  452E  32  9A  4F       13.0       STA   ITERA      ;SAVE COUNT
 (1)  4531  F2  B4  44       10.0       JP    TST04X     ;DO TEST UNTIL TILL = 0
1781
```

```
                                .SBTTL  TEST 05 - LOOP AMTIE TO AMTIE - MTA #0

                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #0
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATH FROM TU PORT #0 TO
                                ;*MTA #0 - #2 AND BACK VIA THE AMTIE LINES.  THROUGH THE MTA #0 AMTIE
                                ;*REGISTER TO THE TU PORT #0 AMTIE REGISTER.
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #0 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 0
                                ;*     CLEAR THE AMTIE DATA
                                ;*     BGNDO
                                ;*     :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #0 AMTIE DATA REGISTER
                                ;*     :   ISSUE THE MSB OF THE AMTIE DATA TO MTA #0 AMTIE 7 DATA BIT
                                ;*     :   ISSUE THE LSB OF THE AMTIE DATA TO MTA #0 AMTIE PARITY REGISTER
                                ;*     :   INPUT THE TU PORT #0 AMTIE DATA REGISTER
                                ;*     :   IF AMTIE DATA READ=AMTIE DATA
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 16
                                ;*     :   ENDIF
                                ;*     :   INPUT THE AMTIE BIT 7 AND PARITY BIT
                                ;*     :   IF THE AMTIE BIT 7 AND PARITY READ=AMTIE BIT 7 AND PARITY WRITTEN
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 17
                                ;*     :   ENDIF
                                ;*     :   INCREMENT THE AMTIE DATA
                                ;*     :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*     ENDDO
                                ;*     CALL SUBROUTINE CLEAR
                                ;*ENDTST
                                SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 05
                                ;*MTA1 MICRO ERROR 16
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #0
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT = DATA READ
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 05
                                        ;*MTA1 MICRO ERROR 17
                                        ;*MTA1-LOOP AMTIE TO AMTIE-MTA #0
                                        ;*M8954, M8955'S
                                        ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4534                                    S
 (1)                                    ; ***************************************************************************
                                        TEST05: TESTX   @05             ;SET UP THE TEST NUMBER
4534                                                    MVI     A,@05           ;DEFINE THE TEST NUMBER
4536  CD  03  28          18.0                         CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP AMTIE TO AMTIE-MTA #0
                                        ;@M8954, M8955'S

4539  3A  CD  49          13.0  TST05X: LDA     UNITMP          ;GET THE UNIT MAP
453C  E6  01                7.0        ANI     @001            ;TEST UNIT 0?
453E  CA  BB  45           10.0        JZ      TEST06          ;NO-CHECK FOR TEST06
4541  CD  A2  49           18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
4544  DB  E0               10.0        IN      INTSTA          ;GET SELECT STATUS
4546  E6  80                7.0        ANI     BIT7
4548  D3  E0               10.0        OUT     MBSEL           ;SELECT PORT 0
454A  21  CE  49           10.0        LXI     H,DATA          ;LOAD ADDRESS OF DATA
454D  AF                    4.0        XRA     A               ;CLEAR THE ACCUMULATOR
454E  77                    7.0        MOV     M,A             ;STORE IN DATA
                                                                ;
454F  3E  80                7.0  ALP01:  MVI     A,@200          ;SELECT MTA #0 REGISTER 0
4551  D3  40               10.0        OUT     TCMD
4553  7E                    7.0        MOV     A,M             ;GET THE AMTIE DATA
4554  E6  01                7.0        ANI     $01             ;GET THE PSEUDO PARITY BIT
4556  47                    4.0        MOV     B,A             ;SAVE IN B
4557  7E                    7.0        MOV     A,M             ;GET THE AMTIE DATA-AGAIN
4558  E6  80                7.0        ANI     @200            ;GET THE MSB
455A  07                    4.0        RLC                     ;POSITION DATA FOR OUTPUT
455B  07                    4.0        RLC
455C  B0                    4.0        ORA     B               ;OR IN THE PSEUDO PARITY BIT
455D  F6  04                7.0        ORI     @004            ;OR IN THE LOOP W/R BIT
455F  D3  40               10.0        OUT     TCMD            ;LOAD MTA REGISTER 0
4561  3E  82                7.0        MVI     A,@202          ;SELECT MTA #0 REGISTER 2
4563  D3  40               10.0        OUT     TCMD
4565  7E                    7.0        MOV     A,M             ;GET THE AMTIE DATA
4566  47                    4.0        MOV     B,A             ;SAVE FOR DATA COMPARE
4567  E6  7F                7.0        ANI     $7F             ;REMOVE THE MSB
4569  D3  40               10.0        OUT     TCMD            ;LOAD MTA REGISTER 2
456B  DB  44               10.0        IN      TAMT            ;READ THE AMTIE REGISTER
456D  B8                    4.0        CMP     B               ;COMPARE ACTUAL AND EXP.
456E  CA  7D  45           10.0        JZ      ALP02           ;EQUAL - CONTINUE
4571                                    ROUT    ADATA           ;STORE ACTUAL DATA
4571  D3  94               10.0                OUT     ADATA           ;WRITE AC INTO ADATA
4573  7F                    4.0                MOV     A,A             ;RETRY LINK
```

```
1874  4574  7E                    7.0          MOV     A,M             ;GET EXP. DATA
1875  4575                                     ROUT    EDATA           ;STORE EXP. DATA
 (1)  4575  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4577  7F                    4.0                  MOV     A,A             ;RETRY LINK
1876  4578                                     ERRB    ALP01,ALP02,0
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4578  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000E                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  457B  0E                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  457C  00                                         .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  457D  CD  15  28            18.0         ALP02:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4580  DA  4F  45            10.0                 JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
1877                                     ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
1878  4583  78                    4.0          MOV     A,B             ;GET EXP. PTY.
1879  4584  E6  01                7.0          ANI     $01             ;REMOVE UNWANTED BITS
1880  4586  47                    4.0          MOV     B,A             ;RESTORE EXP. PTY.
1881  4587  DB  48                10.0         IN      PSTAT           ;GET ACTUAL PTY.
1882  4589  E6  01                7.0          ANI     $01             ;REMOVE UNWANTED BITS
1883  458B  B8                    4.0          CMP     B               ;EQUAL?
1884  458C  CA  9B  45            10.0         JZ      A1LP02          ;YES
1885  458F                                     ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  458F  D3  94                10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4591  7F                    4.0                  MOV     A,A             ;RETRY LINK
1886  4592  78                    4.0          MOV     A,B             ;GET EXP. PTY.
1887  4593                                     ROUT    EDATA           ;STORE EXP. PTY.
 (1)  4593  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4595  7F                    4.0                  MOV     A,A             ;RETRY LINK
1888  4596                                     ERRB    ALP01,A1LP02
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4596  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4599  0F                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459A  00                                         .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  459B  CD  15  28            18.0         A1LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  459E  DA  4F  45            10.0                 JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
1889                                     ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
1890
1891  45A1  7E                    7.0          MOV     A,M             ;GET DATA
1892  45A2  3C                    4.0          INR     A               ;UPDATE THE DATA BYTE
1893  45A3  77                    7.0          MOV     M,A             ;SAVE UPDATED DATA
1894  45A4  C2  4F  45            10.0         JNZ     ALP01           ;CONTINUE UNTIL DONE
1895  45A7                                     ENDTST  TST05X
 (1)                                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45A7                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45A7  CD  06  28            18.0                         CALL    REQST
 (2)  45AA  00                                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  45AB  00  00                                             .WORD                   ;SYSTEM '*' COUNT
 (2)  45AD  00  00                                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45AF  00                                                 .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  45B0  07                                                 .BYTE   7               ;REQUEST CODE
 (1)  45B1  3A  9A  4F            13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  45B4  3D                    4.0          DCR     A               ;DOWNCOUNT
 (1)  45B5  32  9A  4F            13.0         STA     ITERA           ;SAVE COUNT
 (1)  45B8  F2  39  45            10.0         JP      TST05X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 06 - LOOP AMTIE TO AMTIE - MTA #1
45BB                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #1
45BB                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATH FROM TU PORT #1 TO
                                ;*MTA #0 + #2 AND BACK VIA THE AMTIE LINES THROUGH THE MTA #1 AMTIE
                                ;*REGISTER TO THE TU PORT #1 AMTIE REGISTER.
45BB                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT #1 NOT SELECTED BY USER
                                ;*      :   THEN-NEXT TEST
                                ;*      :   ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE CLEAR
                                ;*      SELECT PORT 1
                                ;*      CLEAR THE AMTIE DATA
                                ;*      BGNDO
                                ;*      :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #1 AMTIE DATA REGISTER
                                ;*      :   ISSUE THE MSB OF THE AMTIE DATA TO MTA #1 AMTIE 7 DATA BIT
                                ;*      :   ISSUE THE LSB OF THE AMTIE DATA TO MTA #1 AMTIE PARITY REGISTER
                                ;*      :   INPUT THE TU PORT #1 AMTIE DATA REGISTER
                                ;*      :   IF AMTIE DATA READ=AMTIE DATA
                                ;*      :   :   THEN-CONTINUE
                                ;*      :   :   ELSE-ERROR 20
                                ;*      :   ENDIF
                                ;*      :   INPUT THE AMTIE BIT 7 AND PARITY BIT
                                ;*      :   IF THE AMTIE BIT 7 AND PARITY READ=AMTIE BIT 7 AND PARITY WRITTEN
                                ;*      :   :   THEN-CONTINUE
                                ;*      :   :   ELSE-ERROR 21
                                ;*      :   ENDIF
                                ;*      :   CLEAR TU PORT #1 LOOP W/R BIT
                                ;*      :   INCREMENT THE AMTIE DATA
                                ;*      :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*      ENDDO
                                ;*      CALL SUBROUTINE CLEAR
                                ;*ENDTST
45BB                            SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 06
                                ;*MTA1 MICRO ERROR 20
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #1
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT = DATA READ
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 06
                                        ;*MTA1 MICRO ERROR 21
                                        ;*MTA1-LOOP AMTIE TO AMTIE-MTA #1
                                        ;*M8954, M8955'S
                                        ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
45BB                                    $
 (1)                                    ; ************************************************************************************
                                        TEST06: TESTX   @06             ;SET UP THE TEST NUMBER
45BB                                                    MVI     A,@06           ;DEFINE THE TEST NUMBER
45BB  3E  06                    7.0
45BD  CD  03  28                18.0                    CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP AMTIE TO AMTIE-MTA #1
                                        ;&M8954, M8955'S
45C0  3A  CD  49                13.0    TST06X: LDA     UNITMP          ;GET THE UNIT MAP
45C3  E6  02                    7.0             ANI     @002            ;TEST UNIT 1?
45C5  CA  44  46                10.0            JZ      TEST07          ;NO-CHECK FOR TEST07
45C8  CD  A2  49                18.0            CALL    CLEAR           ;CLEAN UP THE PORTS
45CB  DB  E0                    10.0            IN      INTSTA          ;GET SELECT STATUS
45CD  E6  80                    7.0             ANI     BIT7
45CF  F6  01                    7.0             ORI     1
45D1  D3  E0                    10.0            OUT     MBSEL           ;SELECT PORT 1
45D3  21  CE  49                10.0            LXI     H,DATA          ;LOAD ADDRESS OF DATA
45D6  AF                        4.0             XRA     A               ;CLEAR THE ACCUMULATOR
45D7  77                        7.0             MOV     M,A             ;STORE IN DATA
45D8  3E  80                    7.0     ALP11:  MVI     A,$80           ;SELECT MTA #1 REGISTER 0
45DA  D3  40                    10.0            OUT     TCMD
45DC  7E                        7.0             MOV     A,M             ;GET THE AMTIE DATA
45DD  E6  01                    7.0             ANI     $01             ;GET THE PSEUDO PARITY BIT
45DF  47                        4.0             MOV     B,A             ;SAVE IN B
45E0  7E                        7.0             MOV     A,M             ;GET THE AMTIE DATA-AGAIN
45E1  E6  80                    7.0             ANI     $80             ;GET THE MSB
45E3  07                        4.0             RLC                     ;POSITION DATA FOR OUTPUT
45E4  07                        4.0             RLC
45E5  B0                        4.0             ORA     B               ;OR IN THE PSEUDO PARITY BIT
45E6  F6  04                    7.0             ORI     @004            ;OR IN THE LOOP W/R BIT
45E8  D3  40                    10.0            OUT     TCMD            ;LOAD MTA REGISTER 0
45EA  3E  82                    7.0             MVI     A,$82           ;SELECT MTA #1 REGISTER 2
45EC  D3  40                    10.0            OUT     TCMD
45EE  7E                        7.0             MOV     A,M             ;GET THE AMTIE DATA
45EF  47                        4.0             MOV     B,A             ;SAVE FOR DATA COMPARE
45F0  E6  7F                    7.0             ANI     $7F             ;REMOVE THE MSB
45F2  D3  40                    10.0            OUT     TCMD            ;LOAD MTA REGISTER 2
45F4  DB  44                    10.0            IN      TAMT            ;READ THE AMTIE REGISTER
45F6  B8                        4.0             CMP     B               ;COMPARE ACTUAL AND EXP.
45F7  CA  06  46                10.0            JZ      ALP12           ;EQUAL - CONTINUE
45FA                                            ROUT    ADATA           ;STORE ACTUAL DATA
45FA  D3  94                    10.0    .               OUT     ADATA           ;WRITE AC INTO ADATA
45FC  7F                        4.0     .               MOV     A,A             ;RETRY LINK
```

```
1988  45FD  7E                7.0          MOV     A,M             ;GET EXP. DATA
1989  45FE                                 ROUT    EDATA           ;STORE EXP. DATA
 (1)  45FE  D3  95             10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4600  7F                 4.0                  MOV     A,A             ;RETRY LINK
1990  4601                                 ERRB    ALP11,ALP12,0
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4601  CD  12  28         18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0010                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4604  10                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4605  00                                      .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  4606  CD  15  28         18.0         ALP12:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4609  DA  D8  45         10.0                 JC      ALP11           ;LOOP ADDRESS IF LOOP SPECIFIED
1991                                 ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
1992  460C  78                4.0          MOV     A,B             ;GET EXP. PTY.
1993  460D  E6  01             7.0          ANI     $01             ;REMOVE UNWANTED BITS
1994  460F  47                 4.0          MOV     B,A             ;RESTORE EXP. PTY.
1995  4610  DB  48             10.0         IN      PSTAT           ;GET ACTUAL PTY.
1996  4612  E6  01             7.0          ANI     $01             ;REMOVE UNWANTED BITS
1997  4614  B8                 4.0          CMP     B               ;EQUAL?
1998  4615  CA  24  46         10.0         JZ      A2LP02          ;YES
1999  4618                                 ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  4618  D3  94             10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  461A  7F                 4.0                  MOV     A,A             ;RETRY LINK
2000  461B  78                 4.0          MOV     A,B             ;GET EXP. PTY.
2001  461C                                 ROUT    EDATA           ;STORE EXP. PTY.
 (1)  461C  D3  95             10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  461E  7F                 4.0                  MOV     A,A             ;RETRY LINK
2002  461F                                 ERRB    ALP11,A2LP02
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  461F  CD  12  28         18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0011                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4622  11                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4623  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4624  CD  15  28         18.0         A2LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4627  DA  D8  45         10.0                 JC      ALP11           ;LOOP ADDRESS IF LOOP SPECIFIED
2003                                 ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
2004
2005  462A  7E                7.0          MOV     A,M             ;GET DATA
2006  462B  3C                4.0          INR     A               ;UPDATE THE DATA BYTE
2007  462C  77                7.0          MOV     M,A             ;SAVE UPDATED DATE
2008  462D  C2  D8  45         10.0         JNZ     ALP11           ;CONTINUE UNTIL DONE
2009  4630                                 ENDTST  TST06X
 (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4630                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4630  CD  06  28         18.0                 CALL    REQST
 (2)  4633  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4634  00  00                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  4636  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4638  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4639  07                                      .BYTE   7               ;REQUEST CODE
 (1)  463A  3A  9A  4F         13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  463D  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  463E  32  9A  4F         13.0         STA     ITERA           ;SAVE COUNT
 (1)  4641  F2  C0  45         10.0         JP      TST06X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 07 - LOOP AMTIE TO AMTIE - MTA #2
4644                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #2
4644                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATH FROM TU PORT #2 TO
                                ;*MTA #0 AND #2 AND BACK VIA THE AMTIE LINES THROUGH THE MTA #2 AMTIE
                                ;*REGISTER TO TU PORT 2 AMTIE REGISTER.
4644                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 2
                                ;*    CLEAR THE AMTIE DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #2 AMTIE DATA REGISTER
                                ;*    :   ISSUE THE MSB OF THE AMTIE DATA TO MTA #2 AMTIE 7 DATA BIT
                                ;*    :   ISSUE THE LSB OF THE AMTIE DATA REGISTER TO MTA #2 AMTIE PARITY REGISTER
                                ;*    :   INPUT THE TU PORT #2 AMTIE DATA REGISTER
                                ;*    :   IF AMTIE DATA READ=AMTIE DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR 22
                                ;*    :   ENDIF
                                ;*    :   INPUT THE AMTIE BIT 7 ANDPARITY BIT
                                ;*    :   IF THE AMTIE BIT 7 AND PARITY READ=AMTIE BIT 7 AND PARITY WRITTEN
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR 23
                                ;*    :   ENDIF
                                ;*    :   CLEAR TU PORT #2 LOOP W/R BIT
                                ;*    :   INCREMENT THE AMTIE DATA
                                ;*    :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
4644                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 07
                                ;*MTA1 MICRO ERROR 22
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #2
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT = DATA READ
```

```
2053                                    ;*ACTUAL = NNNN
2054                                    ;*EXPECTED = NNNN
2055                                    ;*
2056                                    ;*MTA1 MICRO TEST 07
2057                                    ;*MTA1 MICRO ERROR 23
2058                                    ;*MTA1-LOOP AMTIE TO AMTIE-MTA #2
2059                                    ;*M8954, M8955'S
2060                                    ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
2061                                    ;*ACTUAL = NNNN
2062                                    ;*EXPECTED = NNNN
2063  4644                              S
 (1)                                    ; ************************************************************************************
2064
2065  4644                              TEST07: TESTX   @07             ;SET UP THE TEST NUMBER
 (1)  4644  3E  07          7.0                         MVI     A,@07           ;DEFINE THE TEST NUMBER
 (1)  4646  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
2066                                    ;%MTA1-LOOP AMTIE TO AMTIE-MTA #2
2067                                    ;&M8954, M8955'S
2068
2069  4649  3A  CD  49     13.0  TST07X: LDA     UNITMP          ;GET THE UNIT MAP
2070  464C  E6  04           7.0        ANI     @004            ;TEST FOR UNIT 2
2071  464E  CA  CD  46      10.0        JZ      TEST10          ;NO-CHECK FOR TEST10
2072  4651  CD  A2  49      18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
2073  4654  DB  E0          10.0        IN      INTSTA          ;GET SELECT STATUS
2074  4656  E6  80           7.0        ANI     BIT7
2075  4658  F6  02           7.0        ORI     2
2076  465A  D3  E0          10.0        OUT     MBSEL           ;SELECT PORT 2
2077  465C  21  CE  49      10.0        LXI     H,DATA          ;LOAD ADDRESS OF DATA
2078  465F  AF                4.0        XRA     A               ;CLEAR THE ACCUMULATOR
2079  4660  77                7.0        MOV     M,A             ;STORE IN DATA
2080  4661  3E  80           7.0  ALP21:  MVI     A,$80           ;SELECT MTA #2 REGISTER 0
2081  4663  D3  40          10.0        OUT     TCMD
2082  4665  7E                7.0        MOV     A,M             ;GET THE AMTIE DATA
2083  4666  E6  01           7.0        ANI     $01             ;GET THE PSEUDO PARITY BIT
2084  4668  47                4.0        MOV     B,A             ;SAVE IN B
2085  4669  7E                7.0        MOV     A,M             ;GET THE AMTIE DATA-AGAIN
2086  466A  E6  80           7.0        ANI     $80             ;GET THE MSB
2087  466C  07                4.0        RLC                     ;POSITION DATA FOR OUTPUT
2088  466D  07                4.0        RLC
2089  466E  B0                4.0        ORA     B               ;OR IN THE PSEUDO PARITY BIT
2090  466F  F6  04           7.0        ORI     @004            ;OR IN THE LOOP W/R BIT
2091  4671  D3  40          10.0        OUT     TCMD            ;LOAD MTA REGISTER 0
2092  4673  3E  82           7.0        MVI     A,$82           ;SELECT MTA #2 REGISTER 2
2093  4675  D3  40          10.0        OUT     TCMD
2094  4677  7E                7.0        MOV     A,M             ;GET THE AMTIE DATA
2095  4678  47                4.0        MOV     B,A             ;SAVE FOR DATA COMPARE
2096  4679  E6  7F           7.0        ANI     $7F             ;REMOVE THE MSB
2097  467B  D3  40          10.0        OUT     TCMD            ;LOAD MTA REGISTER 2
2098  467D  DB  44          10.0        IN      TAMT            ;READ THE AMTIE REGISTER
2099  467F  B8                4.0        CMP     B               ;COMPARE ACTUAL AND EXP.
2100  4680  CA  8F  46      10.0        JZ      ALP22           ;EQUAL - CONTINUE
2101  4683                              ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4683  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4685  7F               4.0                MOV     A,A             ;RETRY LINK
```

```
2102  4686  7E                    7.0          MOV     A,M                 ;GET EXP. DATA
2103  4687                                       ROUT    EDATA               ;STORE EXP. DATA
 (1)  4687  D3  95                10.0                 OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  4689  7F                    4.0                  MOV     A,A                 ;RETRY LINK
2104  468A                                       ERRB    ALP21,ALP22,0
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  468A  CD  12  28            18.0                 CALL    ERLPB               ;PROCESS ERROR - DO 2.3
 (1)        0012                                       MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  468D  12                                         .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  468E  00                                         .BYTE   0                   ;PRINT ROUTINE NUMBER
 (1)  468F  CD  15  28            18.0         ALP22:: CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4692  DA  61  46            10.0                 JC      ALP21               ;LOOP ADDRESS IF LOOP SPECIFIED
2105                                       ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
2106  4695  78                    4.0          MOV     A,B                 ;GET EXP. PTY.
2107  4696  E6  01                7.0          ANI     $01                 ;REMOVE UNWANTED BITS
2108  4698  47                    4.0          MOV     B,A                 ;RESTORE EXP. PTY.
2109  4699  DB  48                10.0         IN      PSTAT               ;GET ACTUAL PTY.
2110  469B  E6  01                7.0          ANI     $01                 ;REMOVE UNWANTED BITS
2111  469D  B8                    4.0          CMP     B                   ;EQUAL?
2112  469E  CA  AD  46            10.0         JZ      A3LP02              ;YES
2113  46A1                                       ROUT    ADATA               ;STORE ACTUAL PTY.
 (1)  46A1  D3  94                10.0                 OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  46A3  7F                    4.0                  MOV     A,A                 ;RETRY LINK
2114  46A4  78                    4.0          MOV     A,B                 ;GET EXP. PTY.
2115  46A5                                       ROUT    EDATA               ;STORE EXP. PTY.
 (1)  46A5  D3  95                10.0                 OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  46A7  7F                    4.0                  MOV     A,A                 ;RETRY LINK
2116  46A8                                       ERRB    ALP21,A3LP02
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46A8  CD  12  28            18.0                 CALL    ERLPB               ;PROCESS ERROR - DO 2.3
 (1)        0013                                       MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46AB  13                                         .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  46AC  00                                         .BYTE                       ;PRINT ROUTINE NUMBER
 (1)  46AD  CD  15  28            18.0         A3LP02::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46B0  DA  61  46            10.0                 JC      ALP21               ;LOOP ADDRESS IF LOOP SPECIFIED
2117                                       ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
2118
2119  46B3  7E                    7.0          MOV     A,M                 ;GET DATA
2120  46B4  3C                    4.0          INR     A                   ;UPDATE THE DATA BYTE
2121  46B5  77                    7.0          MOV     M,A                 ;SAVE UPDATED DATA
2122  46B6  C2  61  46            10.0         JNZ     ALP21               ;CONTINUE UNTIL DONE
2123  46B9                                       ENDTST  TST07X
 (1)                                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46B9                                  REQ     7                   ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46B9  CD  06  28            18.0                 CALL    REQST
 (2)  46BC  00                                         .BYTE                       ;DATA PATTERN NUMBER
 (2)  46BD  00  00                                     .WORD                       ;SYSTEM '*' COUNT
 (2)  46BF  00  00                                     .WORD                       ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46C1  00                                         .BYTE                       ;DATA COMPARE FLAG IF =1
 (2)  46C2  07                                         .BYTE   7                   ;REQUEST CODE
 (1)  46C3  3A  9A  4F            13.0         LDA     ITERA               ;GET ITERATION COUNT
 (1)  46C6  3D                    4.0          DCR     A                   ;DOWNCOUNT
 (1)  46C7  32  9A  4F            13.0         STA     ITERA               ;SAVE COUNT
 (1)  46CA  F2  49  46            10.0         JP      TST07X              ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - LOOP AMTIE TO AMTIE - MTA #3
                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #3
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATH FROM TU PORT #3 TO
                                ;*MTA #0 + #2 AND BACK VIA THE AMTIE LINES THROUGH THE MTA #3 AMTIE
                                ;*REGISTER TO THE TU PORT #3 AMTIE REGISTER.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #3 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 3
                                ;*     CLEAR THE AMTIE DATA
                                ;*     BGNDO
                                ;*     :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #3 AMTIE DATA REGISTER
                                ;*     :   ISSUE THE LSB OF THE AMTIE DATA REGISTER TO MTA #3 AMTIE PARITY REGISTER
                                ;*     :   INPUT THE TU PORT #3 AMTIE DATA REGISTER
                                ;*     :   IF AMTIE DATA READ=AMTIE DATA
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 24
                                ;*     :   ENDIF
                                ;*     :   INPUT THE AMTIE BIT 7 AND PARITY BIT
                                ;*     :   IF THE AMTIE BIT 7 AND PARITY READ=AMTIE BIT 7 AND PARITY WRITTEN
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 25
                                ;*     :   ENDIF
                                ;*     :   CLEAR TU PORT #3 LOOP W/R BIT
                                ;*     :   INCREMENT THE AMTIE DATA
                                ;*     :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*     ENDDO
                                ;*     CALL SUBROUTINE CLEAR
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 10
                                ;*MTA1 MICRO ERROR 24
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #3
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT = DATA READ
                                ;*ACTUAL = NNNN
```

```
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 10
                                        ;*MTA1 MICRO ERROR 25
                                        ;*MTA1-LOOP AMTIE TO AMTIE-MTA #3
                                        ;*M8954, M8955'S
                                        ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        $
                                        ; ***************************************************************************
                                        TEST10: TESTX   @010            ;SET UP THE TEST NUMBER
                                                MVI     A,@010          ;DEFINE THE TEST NUMBER
                                                CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP AMTIE TO AMTIE-MTA #3
                                        ;&M8954, M8955'S

                                        TST10X: LDA     UNITMP          ;GET THE UNIT MAP
                                                ANI     @010            ;TEST FOR UNIT 3?
                                                JZ      TEST11          ;NO-CHECK FOR TEST11
                                                CALL    CLEAR           ;CLEAN UP THE PORTS
                                                IN      INTSTA          ;GET SELECT STATUS
                                                ANI     BIT7
                                                ORI     3
                                                OUT     MBSEL           ;SELECT PORT 3
                                                LXI     H,DATA          ;LOAD ADDRESS OF DATA
                                                XRA     A               ;CLEAR THE ACCUMULATOR
                                                MOV     M,A             ;STORE IN DATA
                                        ALP31:  MVI     A,$80           ;SELECT MTA #3 REGISTER 0
                                                OUT     TCMD
                                                MOV     A,M             ;GET THE AMTIE DATA
                                                ANI     $01             ;GET THE PSEUDO PARITY BIT
                                                MOV     B,A             ;SAVE IN B
                                                MOV     A,M             ;GET THE AMTIE DATA-AGAIN
                                                ANI     $80             ;GET THE MSB
                                                RLC                     ;POSITION DATA FOR OUTPUT
                                                RLC
                                                ORA     B               ;OR IN THE PSEUDO PARITY BIT
                                                ORI     @004            ;OR IN THE LOOP W/R BIT
                                                OUT     TCMD            ;LOAD MTA REGISTER 0
                                                MVI     A,$82           ;SELECT MTA #3 REGISTER 2
                                                OUT     TCMD
                                                MOV     A,M             ;GET THE AMTIE DATA
                                                MOV     B,A             ;SAVE FOR DATA COMPARE
                                                ANI     $7F             ;REMOVE THE MSB
                                                OUT     TCMD            ;LOAD MTA REGISTER 2
                                                IN      TAMT            ;READ THE AMTIE REGISTER
                                                CMP     B               ;COMPARE ACTUAL AND EXP.
                                                JZ      ALP32           ;EQUAL - CONTINUE
                                                ROUT    ADATA           ;STORE ACTUAL DATA
                                                        OUT     ADATA           ;WRITE AC INTO ADATA
                                                        MOV     A,A             ;RETRY LINK
                                                MOV     A,M             ;GET EXP. DATA
```

```
2216  4710                                                  ROUT    EDATA           ;STORE EXP. DATA
 (1)  4710  D3  95          10.0                                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4712  7F               4.0                                   MOV     A,A             ;RETRY LINK
2217  4713                                                  ERRB    ALP31,ALP32,0
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4713  CD  12  28      18.0                                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0014                                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4716  14                                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4717  00                                                     .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  4718  CD  15  28      18.0                           ALP32:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  471B  DA  EA  46      10.0                                   JC      ALP31           ;LOOP ADDRESS IF LOOP SPECIFIED
2218                                                  ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
2219  471E  78                4.0                    MOV     A,B             ;GET EXP. PTY.
2220  471F  E6  01             7.0                    ANI     $01             ;REMOVE UNWANTED BITS
2221  4721  47                 4.0                    MOV     B,A             ;RESTORE EXP. PTY.
2222  4722  DB  48            10.0                    IN      PSTAT           ;GET ACTUAL PTY.
2223  4724  E6  01             7.0                    ANI     $01             ;REMOVE UNWANTED BITS
2224  4726  B8                 4.0                    CMP     B               ;EQUAL?
2225  4727  CA  36  47         10.0                    JZ      A4LP02          ;YES
2226  472A                                            ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  472A  D3  94            10.0                           OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  472C  7F                4.0                            MOV     A,A             ;RETRY LINK
2227  472D  78                 4.0                    MOV     A,B             ;GET EXP. PTY.
2228  472E                                            ROUT    EDATA           ;STORE EXP. PTY.
 (1)  472E  D3  95            10.0                           OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4730  7F                4.0                            MOV     A,A             ;RETRY LINK
2229  4731                                            ERRB    ALP31,A4LP02
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4731  CD  12  28      18.0                           CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0015                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4734  15                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4735  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4736  CD  15  28      18.0                   A4LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4739  DA  EA  46      10.0                           JC      ALP31           ;LOOP ADDRESS IF LOOP SPECIFIED
2230                                          ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
2231
2232  473C  7E                 7.0            MOV     A,M             ;GET DATA
2233  473D  3C                 4.0            INR     A               ;UPDATE THE DATA BYTE
2234  473E  77                 7.0            MOV     M,A             ;SAVE UPDATED DATA
2235  473F  C2  EA  46         10.0           JNZ     ALP31           ;CONTINUE UNTIL DONE
2236
2237  4742                                    ENDTST  TST10X
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4742                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4742  CD  06  28      18.0                           CALL    REQST
 (2)  4745  00                                             .BYTE                   ;DATA PATTERN NUMBER
 (2)  4746  00  00                                         .WORD                   ;SYSTEM '*' COUNT
 (2)  4748  00  00                                         .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  474A  00                                             .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  474B  07                                             .BYTE   7               ;REQUEST CODE
 (1)  474C  3A  9A  4F      13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  474F  3D               4.0           DCR     A               ;DOWNCOUNT
 (1)  4750  32  9A  4F      13.0           STA     ITERA           ;SAVE COUNT
 (1)  4753  F2  D2  46      10.0           JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 11 - LOOP DATA WRITE/READ - MTA #0
4756                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #0
4756                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #0 TO TU
                                ;*PORT #0 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
4756                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN - NEXT TEST
                                ;*    :   ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT TU PORT 0
                                ;*    CLEAR THE DATA BYTE
                                ;*    SET LOOP DATA WRITE/READ MTA 0
                                ;*    BGNDO
                                ;*    :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*    :   OUTPUT THE DATA TO TU PORT #0 COMMAND BYTE
                                ;*    :   INPUT THE DATA FROM TU PORT #0 DATA BYTE
                                ;*    :   IF INPUT DATA = OUTPUT DATA
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 26
                                ;*    :   ENDIF
                                ;*    :   INPUT THE DATA PARITY FROM TU PORT #0
                                ;*    :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 27
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE DATA BYTE
                                ;*    :   DO UNTIL THE DATA = 0
                                ;*    ENDDO
                                ;*ENDTST
4756                            SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 11
                                ;*MTA1 MICRO ERROR 26
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #0
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
2281                                    ;*
2282                                    ;*MTA1 MICRO TEST 11
2283                                    ;*MTA1 MICRO ERROR 27
2284                                    ;*MTA1-LOOP DATA WRITE TO READ-MTA #0
2285                                    ;*M8954, M8955'S
2286                                    ;*DATA PARITY READ NOT = PARITY EXPECTED
2287                                    ;*ACTUAL = NNNN
2288                                    ;*EXPECTED = NNNN
2289                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2290  4756                              S
 (1)                                    ; ***************************************************************************************
2291  4756                              TEST11: TESTX   @11             ;SET UP THE TEST NUMBER
 (1)  4756  3E  09          7.0                     MVI     A,@11       ;DEFINE THE TEST NUMBER
 (1)  4758  CD  03  28     18.0                     CALL    TSET        ;SETUP THE TEST
2292                                    ;%MTA1-LOOP DATA WRITE TO READ-MTA #0
2293                                    ;&M8954, M8955'S
2294
2295  475B  3A  CD  49     13.0  TST11X: LDA     UNITMP                ;GET THE UNIT MAP
2296  475E  E6  01          7.0         ANI     @01                   ;TEST FOR UNIT 0
2297  4760  CA  E6  47     10.0         JZ      TEST12                ;NO-CHECK FOR TEST 12
2298  4763  CD  A2  49     18.0         CALL    CLEAR                 ;CLEAR THE DATA BYTE
2299  4766  DB  E0         10.0         IN      INTSTA                ;GET SELECT STATUS
2300  4768  E6  80          7.0         ANI     BIT7
2301  476A  D3  E0         10.0         OUT     MBSEL                 ;SELECT PORT 0
2302  476C  AF               4.0         XRA     A                     ;
2303  476D  32  CE  49     13.0         STA     DATA                  ;SAVE THE PROGRAM DATA
2304  4770                              ROUT    R05H                  ;CLEAR UPPER BYTE FOR REPORTING
 (1)  4770  D3  88         10.0                 OUT     R05H                ;WRITE AC INTO R05H
 (1)  4772  7F              4.0                 MOV     A,A                 ;RETRY LINK
2305  4773  3E  80          7.0         MVI     A,$80                 ;SELECT MTA #0 REGISTER #0
2306  4775  D3  40         10.0         OUT     TCMD                  ;
2307  4777  3E  04          7.0         MVI     A,$04                 ;SET MTA #0 LOOP WRITE TO READ
2308  4779  D3  40         10.0         OUT     TCMD                  ;
2309  477B  3E  87          7.0  DOLP01: MVI     A,$87                 ;SELECT NON-X-REGISTER
2310  477D  D3  40         10.0         OUT     TCMD
2311  477F  3A  CE  49     13.0         LDA     DATA                  ;GET THE DATA BYTE
2312  4782  D3  40         10.0         OUT     TCMD                  ;WRITE TO THE PORT
2313  4784                              ROUT    R05L                  ;AND AGAIN
 (1)  4784  D3  8A         10.0                 OUT     R05L                ;WRITE AC INTO R05L
 (1)  4786  7F              4.0                 MOV     A,A                 ;RETRY LINK
2314  4787                              ROUT    EDATA                 ;SAVE IN THE CAS
 (1)  4787  D3  95         10.0                 OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  4789  7F              4.0                 MOV     A,A                 ;RETRY LINK
2315  478A  47               4.0         MOV     B,A                   ;SAVE IN REGISTER B
2316  478B  DB  4C         10.0         IN      PRDD                  ;GET THE ACTUAL DATA
2317  478D                              ROUT    ADATA                 ;SAVE THE ACTUAL DATA IN CAS
 (1)  478D  D3  ?          10.0                 OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  478F  7F              4.0                 MOV     A,A                 ;RETRY LINK
2318  4790  B8               4.0         CMP     B                     ;ACTUAL DATA = EXPECTED DATA?
2319  4791  CA  99  47     10.0         JZ      DOLP02                ;YES - CONTINUE
```

```
2321  4794                                                ERRB    DOLP01,DOLP02,0
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4794  CD  12  28        18.0                              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0016                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4797  16                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4798  00                                                .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  4799  CD  15  28        18.0                      DOLP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  479C  DA  7B  47        10.0                              JC      DOLP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2322                                                    ;>DATA WRITTEN NOT = DATA READ
2323  479F  3A  CE  49        13.0                      LDA     DATA            ;GET THE EXPECTED DATA
2324  47A2  A7                  4.0                      ANA     A
2325  47A3  EA  AA  47        10.0                      JPE     DOLPP1          ;SET EXPECTED PARITY
2326  47A6  AF                  4.0              DOLPP0: XRA     A               ;EXPECTED PARITY =0
2327  47A7  C3  AC  47        10.0                      JMP     DOLPPC
2328  47AA  3E  01              7.0              DOLPP1: MVI     A,1             ;EXPECTED PARITY =1
2329  47AC                                       DOLPPC: ROUT    EDATA           ;SAVE IT
 (1)  47AC  D3  95            10.0                              OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47AE  7F                 4.0                              MOV     A,A             ;RETRY LINK
2330  47AF  47                  4.0                      MOV     B,A             ;SAVE IN REGISTER B
2331  47B0  DB  48             10.0                      IN      PSTAT           ;GET THE ACTUAL PARITY
2332  47B2  E6  02              7.0                      ANI     P.RPST          ;REMOVE ONLY THE PARITY BIT
2333  47B4  0F                  4.0                      RRC                     ;POSITION THE BIT
2334  47B5                                               ROUT    ADATA           ;STORE IN CAS
 (1)  47B5  D3  94            10.0                              OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47B7  7F                 4.0                              MOV     A,A             ;RETRY LINK
2335  47B8  B8                  4.0                      CMP     B               ;ACTUAL PARITY = EXPECTED PARITY?
2336  47B9  CA  C1  47         10.0                      JZ      DOLP03          ;YES - CONTINUE
2337  47BC                                               ERRB    DOLP01,DOLP03,1
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  47BC  CD  12  28        18.0                              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0017                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47BF  17                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47C0  01                                                .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  47C1  CD  15  28        18.0                      DOLP03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47C4  DA  7B  47        10.0                              JC      DOLP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2338                                                    ;>DATA PARITY READ NOT = PARITY EXPECTED
2339  47C7  3A  CE  49        13.0                      LDA     DATA            ;GET THE DATA BYTE
2340  47CA  3C                  4.0                      INR     A               ;INCREMENT
2341  47CB  32  CE  49        13.0                      STA     DATA            ;SAVE THE NEW DATA
2342  47CE  A7                  4.0                      ANA     A               ;SET THE CONDITION BITS
2343  47CF  C2  7B  47         10.0                      JNZ     DOLP01
2344  47D2                                               ENDTST  TST11X
 (1)                                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47D2                                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47D2  CD  06  28        18.0                              CALL    REQST
 (2)  47D5  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  47D6  00  00                                            .WORD                   ;SYSTEM '*' COUNT
 (2)  47D8  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47DA  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  47DB  07                                                .BYTE   7               ;REQUEST CODE
 (1)  47DC  3A  9A  4F        13.0                      LDA     ITERA           ;GET ITERATION COUNT
 (1)  47DF  3D                 4.0                      DCR     A               ;DOWNCOUNT
 (1)  47E0  32  9A  4F        13.0                      STA     ITERA           ;SAVE COUNT
 (1)  47E3  F2  5B  47        10.0                      JP      TST11X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - LOOP DATA WRITE/READ - MTA #1
47E6                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #1
47E6                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #1 TO TU
                                ;*PORT #1 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
47E6                            SP
                                ;******************   *********************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #1 NOT SELECTED BY USER
                                ;*     :   THEN - NEXT TEST
                                ;*     :   ELSE - CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT TU PORT 1
                                ;*     CLEAR THE DATA BYTE
                                ;*     SET LOOP DATA WRITE/READ MTA 1
                                ;*     BGNDO
                                ;*     :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*     :   OUTPUT THE DATA TO TU PORT #1 COMMAND BYTE
                                ;*     :   INPUT THE DATA FROM TU PORT #1 DATA BYTE
                                ;*     :   IF INPUT DATA = OUTPUT DATA
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 30
                                ;*     :   ENDIF
                                ;*     :   INPUT THE DATA PARITY FROM TU PORT #1
                                ;*     :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 31
                                ;*     :   ENDIF
                                ;*     :   INCREMEN  THE DATA BYTE
                                ;*     :   DO UNTIL THE DATA = 0
                                ;*     ENDDO
                                ;*ENDTST
47E6                            SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 12
                                ;*MTA1 MICRO ERROR 30
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #1
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
2388                                    ;*
2389                                    ;*MTA1 MICRO TEST 12
2390                                    ;*MTA1 MICRO ERROR 31
2391                                    ;*MTA1-LOOP DATA WRITE TO READ-MTA #1
2392                                    ;*M8954, M8955'S
2393                                    ;*DATA PARITY READ NOT = PARITY EXPECTED
2394                                    ;*ACTUAL = NNNN
2395                                    ;*EXPECTED = NNNN
2396  47E6                              S
 (1)                                    ; *****************************************************************************
2397  47E6                              TEST12: TESTX   @12             ;SET UP THE TEST NUMBER
 (1)  47E6  3E  0A            7.0                    MVI     A,@12           ;DEFINE THE TEST NUMBER
 (1)  47E8  CD  03  28       18.0                    CALL    TSET            ;SETUP THE TEST
2398                                    ;%MTA1-LOOP DATA WRITE TO READ-MTA #1
2399                                    ;&M8954, M8955'S
2400
2401  47EB  3A  CD  49         13.0     TST12X: LDA     UNITMP          ;GET THE UNIT MAP
2402  47EE  E6  02              7.0             ANI     @02             ;TEST FOR UNIT 1
2403  47F0  CA  78  48         10.0             JZ      TEST13          ;NO-CHECK FOR TEST 13
2404  47F3  CD  A2  49         18.0             CALL    CLEAR           ;CLEAR THE DATA BYTE
2405  47F6  DB  E0             10.0             IN      INTSTA          ;GET SELECT STATUS
2406  47F8  E6  80              7.0             ANI     BIT7
2407  47FA  F6  01              7.0             ORI     1
2408  47FC  D3  E0             10.0             OUT     MBSEL           ;SELECT PORT 1
2409  47FE  AF                  4.0             XRA     A               ;
2410  47FF  32  CE  49         13.0             STA     DATA            ;SAVE THE PROGRAM DATA
2411  4802                              ROUT    R05H
 (1)  4802  D3  88             10.0                    OUT     R05H            ;WRITE AC INTO R05H
 (1)  4804  7F                  4.0                    MOV     A,A             ;RETRY LINK
2412  4805  3E  80              7.0             MVI     A,$80           ;SELECT MTA #1 REGISTER #0
2413  4807  D3  40             10.0             OUT     TCMD            ;
2414  4809  3E  04              7.0             MVI     A,$04           ;SET MTA #1 LOOP WRITE TO READ
2415  480B  D3  40             10.0             OUT     TCMD            ;
2416  480D  3E  87              7.0     D1LP01: MVI     A,$87           ;SELECT NON-X-REG
2417  480F  D3  40             10.0             OUT     TCMD
2418  4811  3A  CE  49         13.0             LDA     DATA            ;GET THE DATA BYTE
2419  4814  D3  40             10.0             OUT     TCMD            ;WRITE TO THE PORT
2420  4816                              ROUT    R05L            ;SAVE FOR REPORTING
 (1)  4816  D3  8A             10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  4818  7F                  4.0                    MOV     A,A             ;RETRY LINK
2421  4819                              ROUT    EDATA           ;SAVE IN THE CAS
 (1)  4819  D3  95             10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  481B  7F                  4.0                    MOV     A,A             ;RETRY LINK
2422  481C  47                  4.0             MOV     B,A             ;SAVE IN REGISTER B
2423  481D  DB  4C             10.0             IN      PRDD            ;GET THE ACTUAL DATA
2424  481F                              ROUT    ADATA           ;SAVE THE ACTUAL DATA IN CAS
 (1)  481F  D3  94             10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4821  7F                  4.0                    MOV     A,A             ;RETRY LINK
2425  4822  B8                  4.0             CMP     B               ;ACTUAL DATA = EXPECTED DATA?
2426  4823  CA  2B  48         10.0             JZ      D1LP02          ;YES - CONTINUE
```

```
                                        ERRB    D1LP01,D1LP02,0
                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE   0               ;PRINT ROUTINE NUMBER
                                        D1LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      D1LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>DATA WRITTEN NOT = DATA READ
                                        LDA     DATA            ;GET THE EXPECTED DATA
                                        ANA     A
                                        JPE     D1LPP1          ;SET EXPECTED PARITY
                                D1LPP0: XRA     A               ;EXPECTED PARITY =0
                                        JMP     D1LPPC
                                D1LPP1: MVI     A,1             ;EXPECTED PARITY =1
                                D1LPPC: ROUT    EDATA           ;SAVE IT
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        MOV     B,A             ;SAVE IN REGISTER B
                                        IN      PSTAT           ;GET THE ACTUAL PARITY
                                        ANI     P.RPST          ;REMOVE ONLY THE PARITY BIT
                                        RRC                     ;POSITION THE BIT
                                        ROUT    ADATA           ;STORE IN CAS
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        CMP     B               ;ACTUAL PARITY = EXPECTED PARITY?
                                        JZ      D1LP03          ;YES - CONTINUE
                                        ERRB    D1LP01,D1LP03,1
                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE   1               ;PRINT ROUTINE NUMBER
                                        D1LP03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      D1LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>DATA PARITY READ NOT = PARITY EXPECTED
                                        LDA     DATA            ;GET THE DATA BYTE
                                        INR     A               ;INCREMENT
                                        STA     DATA            ;SAVE THE NEW DATA
                                        ANA     A               ;SET THE CONDITION BITS
                                        JNZ     D1LP01
                                        ENDTST  TST12X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
                                                CALL    REQST
                                                .BYTE                   ;DATA PATTERN NUMBER
                                                .WORD                   ;SYSTEM "#" COUNT
                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                .BYTE                   ;DATA COMPARE FLAG IF =1
                                                .BYTE   7               ;REQUEST CODE
                                        LDA     ITERA           ;GET ITERATION COUNT
                                        DCR     A               ;DOWNCOUNT
                                        STA     ITERA           ;SAVE COUNT
                                        JP      TST12X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - LOOP DATA WRITE/READ - MTA #2
4878                            ST
                                ;*********************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #2
4878                            SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #2 TO TU
                                ;*PORT #2 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
4878                            SP
                                ;*********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #2 NOT SELECTED BY USER
                                ;*     :   THEN - NEXT TEST
                                ;*     :   ELSE - CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT TU PORT 2
                                ;*     CLEAR THE DATA BYTE
                                ;*     SET LOOP DATA WRITE/READ MTA 2
                                ;*     BGNDO
                                ;*     :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*     :   OUTPUT THE DATA TO TU PORT #2 COMMAND BYTE
                                ;*     :   INPUT THE DATA FROM TU PORT #2 DATA BYTE
                                ;*     :   IF INPUT DATA = OUTPUT DATA
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 32
                                ;*     :   ENDIF
                                ;*     :   INPUT THE DATA PARITY FROM TU PORT #2
                                ;*     :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 33
                                ;*     :   ENDIF
                                ;*     :   INCREMENT THE DATA BYTE
                                ;*     :   DO UNTIL THE DATA = 0
                                ;*     ENDDO
                                ;*ENDTST
4878                            SE
                                ;*********************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 13
                                ;*MTA1 MICRO ERROR 32
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #2
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED - NNNN
```

```
2495                                    ;*
2496                                    ;*MTA1 MICRO TEST 13
2497                                    ;*MTA1 MICRO ERROR 33
2498                                    ;*MTA1-LOOP DATA WRITE TO READ-MTA #2
2499                                    ;*M8954, M8955'S
2500                                    ;*DATA PARITY READ NOT = PARITY EXPECTED
2501                                    ;*ACTUAL = NNNN
2502                                    ;*EXPECTED = NNNN
2503  4878                              S
 (1)                                    ; ************************************************************************************
2504  4878                              TEST13: TESTX   @13             ;SET UP THE TEST NUMBER
 (1)  4878  3E  0B          7.0                     MVI     A,@13       ;DEFINE THE TEST NUMBER
 (1)  487A  CD  03  28     18.0                     CALL    TSET        ;SETUP THE TEST
2505                                    ;%MTA1-LOOP DATA WRITE TO READ-MTA #2
2506                                    ;&M8954, M8955'S
2507
2508  487D  3A  CD  49     13.0  TST13X: LDA     UNITMP              ;GET THE UNIT MAP
2509  4880  E6  04          7.0        ANI     @04                 ;TEST FOR UNIT 2
2510  4882  CA  0A  49      10.0        JZ      TEST14              ;NO-CHECK FOR TEST 14
2511  4885  CD  A2  49      18.0        CALL    CLEAR               ;CLEAR THE DATA BYTE
2512  4888  DB  E0          10.0        IN      INTSTA              ;GET SELECT STATUS
2513  488A  E6  80          7.0        ANI     BIT7
2514  488C  F6  02          7.0        ORI     2
2515  488E  D3  E0          10.0        OUT     MBSEL               ;SELECT PORT 2
2516  4890  AF              4.0        XRA     A                   ;
2517  4891  32  CE  49      13.0        STA     DATA                ;SAVE THE PROGRAM DATA
2518  4894                              ROUT    R05H
 (1)  4894  D3  88          10.0                OUT     R05H            ;WRITE AC INTO R05H
 (1)  4896  7F              4.0                 MOV     A,A             ;RETRY LINK
2519  4897  3E  80          7.0        MVI     A,$80               ;SELECT MTA #2 REGISTER #0
2520  4899  D3  40          10.0        OUT     TCMD                ;
2521  489B  3E  04          7.0        MVI     A,$04               ;SET MTA #2 LOOP WRITE TO READ
2522  489D  D3  40          10.0        OUT     TCMD                ;
2523  489F  3E  87          7.0  D2LP01: MVI     A,$87               ;GET A NON-X-REGISTER
2524  48A1  D3  40          10.0        OUT     TCMD                ;SELECT IT
2525  48A3  3A  CE  49      13.0        LDA     DATA                ;GET THE DATA BYTE
2526  48A6  D3  40          10.0        OUT     TCMD                ;WRITE TO THE PORT
2527  48A8                              ROUT    R05L                ;SAVE FOR REPORTING
 (1)  48A8  D3  8A          10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  48AA  7F              4.0                 MOV     A,A             ;RETRY LINK
2528  48AB                              ROUT    EDATA               ;SAVE IN THE CAS
 (1)  48AB  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  48AD  7F              4.0                 MOV     A,A             ;RETRY LINK
2529  48AE  47              4.0        MOV     B,A                 ;SAVE IN REGISTER B
2530  48AF  DB  4C          10.0        IN      PRDD                ;GET THE ACTUAL DATA
2531  48B1                              ROUT    ADATA               ;SAVE THE ACTUAL DATA IN CAS
 (1)  48B1  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  48B3  7F              4.0                 MOV     A,A             ;RETRY LINK
2532  48B4  B8              4.0        CMP     B                   ;ACTUAL DATA = EXPECTED DATA?
2533  48B5  CA  BD  48      10.0        JZ      D2LP02              ;YES - CONTINUE
```

```
2535  48B8                                                ERRB    D2LP01,D2LP02,0
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  48B8  CD  12  28        18.0                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001A                                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48BB  1A                                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48BC  00                                                 .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  48BD  CD  15  28        18.0            D2LP02::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48C0  DA  9F  48        10.0                             JC      D2LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2536                                        ;>DATA WRITTEN NOT = DATA READ
2537  48C3  3A  CE  49        13.0                     LDA     DATA            ;GET THE EXPECTED DATA
2538  48C6  A7                  4.0                     ANA     A
2539  48C7  EA  CE  48        10.0                     JPE     D2LPP1          ;SET EXPECTED PARITY
2540  48CA  AF                  4.0             D2LPP0: XRA     A               ;EXPECTED PARITY =0
2541  48CB  C3  D0  48        10.0                     JMP     D2LPPC
2542  48CE  3E  01              7.0             D2LPP1: MVI     A,1             ;EXPECTED PARITY =1
2543  48D0                                      D2LPPC: ROUT    EDATA           ;SAVE IT
 (1)  48D0  D3  95            10.0                             OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  48D2  7F                 4.0                             MOV     A,A             ;RETRY LINK
2544  48D3  47                 4.0                     MOV     B,A             ;SAVE IN REGISTER B
2545  48D4  DB  48             10.0                     IN      PSTAT           ;GET THE ACTUAL PARITY
2546  48D6  E6  02              7.0                     ANI     P.RPST          ;REMOVE ONLY THE PARITY BIT
2547  48D8  0F                  4.0                     RRC                     ;POSITION THE BIT
2548  48D9                                              ROUT    ADATA           ;STORE IN CAS
 (1)  48D9  D3  94            10.0                             OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  48DB  7F                 4.0                             MOV     A,A             ;RETRY LINK
2549  48DC  B8                 4.0                     CMP     B               ;ACTUAL PARITY = EXPECTED PARITY?
2550  48DD  CA  E5  48         10.0                     JZ      D2LP03          ;YES - CONTINUE
2551  48E0                                              ERRB    D2LP01,D2LP03,1
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  48E0  CD  12  28        18.0                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001B                                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48E3  1B                                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48E4  01                                                 .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  48E5  CD  15  28        18.0            D2LP03.:         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48E8  DA  9F  48        10.0                             JC      D2LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2552                                        ;>DATA PARITY READ NOT = PARITY EXPECTED
2553  48EB  3A  CE  49        13.0                     LDA     DATA            ;GET THE DATA BYTE
2554  48EE  3C                  4.0                     INR     A               ;INCREMENT
2555  48EF  32  CE  49        13.0                     STA     DATA            ;SAVE THE NEW DATA
2556  48F2  A7                  4.0                     ANA     A               ;SET THE CONDITION BITS
2557  48F3  C2  9F  48        10.0                     JNZ     D2LP01
2558  48F6                                              ENDTST  TST13X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  48F6                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  48F6  CD  06  28        18.0                                CALL    REQST
 (2)  48F9  00                                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  48FA  00  00                                             .WORD                   ;SYSTEM '*' COUNT
 (2)  48FC  00  00                                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  48FE  00                                                 .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  48FF  07                                                 .BYTE   7               ;REQUEST CODE
 (1)  4900  3A  9A  4F        13.0                     LDA     ITERA           ;GET ITERATION COUNT
 (1)  4903  3D                 4.0                     DCR     A               ;DOWNCOUNT
 (1)  4904  32  9A  4F        13:0                     STA     ITERA           ;SAVE COUNT
 (1)  4907  F2  7D  48        10.0                     JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - LOOP DATA WRITE/READ - MTA #3
490A                            ST
                                ;*************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #3
490A                            SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #3 TO TU
                                ;*PORT #3 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
490A                            SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #3 NOT SELECTED BY USER
                                ;*     :   THEN - NEXT TEST
                                ;*     :   ELSE - CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT TU PORT 3
                                ;*     CLEAR THE DATA BYTE
                                ;*     SET LOOP DATA WRITE/READ MTA 3
                                ;*     BGNDO
                                ;*     :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*     :   OUTPUT THE DATA TO TU PORT #3 COMMAND BYTE
                                ;*     :   INPUT THE DATA FROM TU PORT #3 DATA BYTE
                                ;*     :   IF INPUT DATA = OUTPUT DATA
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 34
                                ;*     :   ENDIF
                                ;*     :   INPUT THE DATA PARITY FROM TU PORT #3
                                ;*     :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 35
                                ;*     :   ENDIF
                                ;*     :   INCREMENT THE DATA BYTE
                                ;*     :   DO UNTIL THE DATA = 0
                                ;*     ENDDO
                                ;*ENDTST
490A                            SE
                                ;*************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 14
                                ;*MTA1 MICRO ERROR 34
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #3
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*MTA1 MICRO TEST 14
                                        ;*MTA1 MICRO ERROR 35
                                        ;*MTA1-LOOP DATA WRITE TO READ-MTA #3
                                        ;*M8954, M8955'S
                                        ;*DATA PARITY READ NOT = PARITY EXPECTED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
490A                                    S
 (1)                                    ; ***************************************************************************
490A                                    TEST14: TESTX   @14             ;SET UP THE TEST NUMBER
 (1) 490A 3E 0C              7.0                        MVI     A,@14       ;DEFINE THE TEST NUMBER
 (1) 490C CD 03 28          18.0                        CALL    TSET        ;SETUP THE TEST
                                        ;%MTA1-LOOP DATA WRITE TO READ-MTA #3
                                        ;@M8954, M8955'S

490F 3A CD 49              13.0 TST14X: LDA     UNITMP          ;GET THE UNIT MAP
4912 E6 08                  7.0          ANI     @10             ;TEST FOR UNIT 3
4914 CA 9C 49              10.0          JZ      EXIT            ;NO-EXIT THE PROGRAM
4917 CD A2 49              18.0          CALL    CLEAR           ;CLEAR THE DATA BYTE
491A DB E0                 10.0          IN      INTSTA          ;GET SELECT STATUS
491C E6 80                  7.0          ANI     BIT7
491E F6 03                  7.0          ORI     3
4920 D3 E0                 10.0          OUT     MBSEL           ;SELECT PORT 3
4922 AF                     4.0          XRA     A               ;
4923 32 CE 49              13.0          STA     DATA            ;SAVE THE PROGRAM DATA
4926                                     ROUT    R05H
 (1) 4926 D3 8B             10.0                    OUT     R05H            ;WRITE AC INTO R05H
 (1) 4928 7F                 4.0                    MOV     A,A             ;RETRY LINK
4929 3E 80                  7.0          MVI     A,$80           ;SELECT MTA #3 REGISTER #0
492B D3 40                 10.0          OUT     TCMD            ;
492D 3E 04                  7.0          MVI     A,$04           ;SET MTA #3 LOOP WRITE TO READ
492F D3 40                 10.0          OUT     TCMD            ;
4931 3E 87                  7.0 D3LP01:  MVI     A,$87           ;SELECT NON-X-REG
4933 D3 40                 10.0          OUT     TCMD
4935 3A CE 49              13.0          LDA     DATA            ;GET THE DATA BYTE
4938 D3 40                 10.0          OUT     TCMD            ;WRITE TO THE PORT
493A                                     ROUT    R05L            ;SAVE FOR REPORTING
 (1) 493A D3 8A             10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1) 493C 7F                 4.0                    MOV     A,A             ;RETRY LINK
493D                                     ROUT    EDATA           ;SAVE IN THE CAS
 (1) 493D D3 95             10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1) 493F 7F                 4.0                    MOV     A,A             ;RETRY LINK
494C 47                     4.0          MOV     B,A             ;SAVE IN REGISTER B
4941 DB 4C                 10.0          IN      PRDD            ;GET THE ACTUAL DATA
4943                                     ROUT    ADATA           ;SAVE THE ACTUAL DATA IN CAS
 (1) 4943 D3 94             10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4945 7F                 4.0                    MOV     A,A             ;RETRY LINK
4946 B8                     4.0          CMP     B               ;ACTUAL DATA = EXPECTED DATA?
4947 CA 4F 49              10.0          JZ      D3LP02          ;YES - CONTINUE
```

```
2642  494A                                        ERRB    D3LP01,D3LP02,0
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  494A  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001C                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  494D  1C                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  494E  00                                            .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  494F  CD  15  28        18.0                D3LP02::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4952  DA  31  49        10.0                        JC      D3LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2643                                    ;>DATA WRITTEN NOT = DATA READ
2644
2645  4955  3A  CE  49        13.0            LDA     DATA            ;GET THE EXPECTED DATA
2646  4958  A7                 4.0             ANA     A
2647  4959  EA  60  49        10.0            JPE     D3LPP1          ;SET EXPECTED PARITY
2648  495C  AF                 4.0     D3LPP0: XRA     A               ;EXPECTED PARITY =0
2649  495D  C3  62  49        10.0            JMP     D3LPPC
2650  4960  3E  01             7.0     D3LPP1: MVI     A,1             ;EXPECTED PARITY =1
2651  4962                             D3LPPC: ROUT    EDATA           ;SAVE IT
 (1)  4962  D3  95             10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4964  7F                  4.0                     MOV     A,A             ;RETRY LINK
2652  4965  47                 4.0             MOV     B,A             ;SAVE IN REGISTER B
2653  4966  DB  48             10.0            IN      PSTAT           ;GET THE ACTUAL PARITY
2654  4968  E6  02             7.0             ANI     P.RPST          ;REMOVE ONLY THE PARITY BIT
2655  496A  0F                 4.0             RRC                     ;POSITION THE BIT
2656  496B                                     ROUT    ADATA           ;STORE IN CAS
 (1)  496B  D3  94             10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  496D  7F                  4.0                     MOV     A,A             ;RETRY LINK
2657  496E  B8                 4.0             CMP     B               ;ACTUAL PARITY = EXPECTED PARITY?
2658  496F  CA  77  49        10.0            JZ      D3LP03          ;YES - CONTINUE
2659  4972                                     ERRB    D3LP01,D3LP03,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4972  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001D                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4975  1D                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4976  01                                            .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4977  CD  15  28        18.0                D3LP03::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  497A  DA  31  49        10.0                        JC      D3LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2660                                    ;>DATA PARITY READ NOT = PARITY EXPECTED
2661  497D  3A  CE  49        13.0            LDA     DATA            ;GET THE DATA BYTE
2662  4980  3C                 4.0             INR     A               ;INCREMENT
2663  4981  32  CE  49        13.0            STA     DATA            ;SAVE THE NEW DATA
2664  4984  A7                 4.0             ANA     A               ;SET THE CONDITION BITS
2665  4985  C2  31  49        10.0            JNZ     D3LP01
```

```
2667  4988                                    ENDTST  TST14X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4988                          REQ     7                    ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4988  CD  06  28      18.0                        CALL    REQST
 (2)  498B  00                                          .BYTE                ;DATA PATTERN NUMBER
 (2)  498C  00  00                                      .WORD                ;SYSTEM '*' COUNT
 (2)  498E  00  00                                      .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4990  00                                          .BYTE                ;DATA COMPARE FLAG IF =1
 (2)  4991  07                                          .BYTE   7            ;REQUEST CODE
 (1)  4992  3A  9A  4F      13.0            LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4995  3D               4.0            DCR     A                       ;DOWNCOUNT
 (1)  4996  32  9A  4F      13.0            STA     ITERA                   ;SAVE COUNT
 (1)  4999  F2  0F  49      10.0            JP      TST14X                  ;DO TEST UNTIL TILL = 0
2668  499C  CD  A2  49      18.0  EXIT:     CALL    CLEAR                   ;LEAVE THE PORTS CLEAN
2669  499F  C3  18  28      10.0            JMP     TSTEND                  ;END OF TEST
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                SSUB
                                ;*********************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CLEAR ALL TU PORTS
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*     CLEAR PORT SELECT FOR TRANSPORT
                                ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                ;*     CLEAR PORT AMTIE WORD
                                ;*ENDSUB
                                S
                                ;*********************************************************************

                                12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                                12.0          PUSH    B
                                 7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
                                10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
                                 7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                                 4.0          ORA     B               ;ADD IN THE SELECTED PORT #
                                10.0          OUT     MBSEL           ;RESET TO THIS PORT
                                 7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                                10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                                 4.0          XRA     A               ;CLEAR TU COMMAND A
                                10.0          OUT     TCMD            ;
                                 7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                                10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                                 7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
                                10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
                                 4.0          XRA     A
                                10.0          OUT     TAMT            ;CLEAR AMTIE WORD
                                10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                                10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
                                 4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                                 4.0          MOV     A,B
                                 7.0          CPI     4               ;DONE?
                                10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                                10.0          POP     B               ;RESET B & C
                                10.0          POP     PSW             ;ALL DONE
                                10.0          RET                     ;EXIT
```

```
2688
2689  49A2  F5                 12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
2690  49A3  C5                 12.0          PUSH    B
2691  49A4  06  00              7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
2692  49A6  DB  E0             10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
2693  49A8  E6  80              7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
2694  49AA  B0                  4.0          ORA     B               ;ADD IN THE SELECTED PORT #
2695  49AB  D3  E0             10.0          OUT     MBSEL           ;RESET TO THIS PORT
2696  49AD  3E  80              7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
2697  49AF  D3  40             10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
2698  49B1  AF                  4.0          XRA     A               ;CLEAR TU COMMAND A
2699  49B2  D3  40             10.0          OUT     TCMD            ;
2700  49B4  3E  81              7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
2701  49B6  D3  40             10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
2702  49B8  3E  00              7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
2703  49BA  D3  40             10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
2704  49BC  AF                  4.0          XRA     A
2705  49BD  D3  44             10.0          OUT     TAMT            ;CLEAR AMTIE WORD
2706  49BF  D3  48             10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
2707  49C1  D3  4C             10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
2708  49C3  04                  4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
2709  49C4  78                  4.0          MOV     A,B
2710  49C5  FE  04              7.0          CPI     4               ;DONE?
2711  49C7  C2  A6  49         10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
2712  49CA  C1                 10.0          POP     B               ;RESET B & C
2713  49CB  F1                 10.0          POP     PSW             ;ALL DONE
2714  49CC  C9                 10.0          RET                     ;EXIT
```

```
                                        .SBTTL  PROGRAM VARIABLES

2716
2717
2718  49CD   00                         UNITMP: .BYTE   0               ;UNIT MAP
2719  49CE   00                         DATA:   .BYTE   0               ;DATA PATTERN
2720         0000                       .END
```

```
A      =%0007          ADATA  = 0094          ALP01  454F          ALP02  457D  G
ALP11  45D8            ALP12  4606  G        ALP21  4661          ALP22  468F  G
ALP31  46EA            ALP32  4718  G        AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE  4F98           ATTCD  4F97          AXNUM  4F91
A1LP02 4598  G         A2LP02 4624  G        A3LP02 46AD  G       A4LP02 4736  G
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE  4F9C           BYTCNT= 00D4          BYTEH  4F24
BYTEL  4F23            C      =%0001          CASCT  4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 0088          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CH0TIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLEAR  49A2            CLKCTL= 00F0           CLOCK  4F26          CLRLP  49A6
CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAVE  4F9D            CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082          C.     = 0001
C.AVAI= 0080           C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002          D      =%0002          DATA   49CE
DATACT= 00D0           DBUS   4F28           DBUSCT= 00C0          DBUSST= 00C0
DDRA   = 00D8          DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002
DDRC   = 00DA          DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB
DIAFLG 4F22            DIAGPG= 4300           DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010           DPEN00 434A           DPEN01 436C  G
DPEN02 4372            DPEN03 438B  G         DPEN10 43CF           DPEN11 43F1  G
DPEN12 43F7            DPEN13 4410  G         DPEN20 4454           DPEN21 4476  G
DPEN22 447C            DPEN23 4495  G         DPEN30 44D9           DPEN31 44FB  G
DPEN32 4501            DPEN33 451A  G         DSAVE  4F9E           DSE    = 0006
DUMMY  431E  G         D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          D0LPPC 47AC           D0LPP0 47A6           D0LPP1 47AA
D0LP01 477B            D0LP02 4799  G         D0LP03 47C1  G        D1LPPC 483E
D1LPP0 4838            D1LPP1 483C           D1LP01 480D           D1LP02 482B  G
D1LP03 4853  G         D2LPPC 48D0           D2LPP0 48CA           D2LPP1 48CE
D2LP01 489F            D2LP02 48BD  G         D2LP03 48E5  G        D3LPPC 4962
D3LPP0 495C            D3LPP1 4960           D3LP01 4931           D3LP02 494F  G
D3LP03 4977  G         E      =%0003          ECCBAD= 0042          ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095
EOTCLR= 0003           ERFLG  4F93           ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM  4F90          ERRCNT= 00D6
ESAVE  4F9F            EXIT   499C           E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A
FORMAT 4F25            FWDTST= 0061           GCRID  = 0089          GCRSET= 0002
```

```
GOODTM= 0092        H     =%0004    HLSAVE  4FA0      IE     = 0008
INTSTA= 00E0        ITERA   4F9A    I.PWR = 0020      I.RMPE= 0040
I5.5  = 0010        I6.5  = 0020    I7.5  = 0040      KCALL = 005F
KCLR  = 007B        KDEP  = 003F    KENAB = 0078      KEXAM = 003E
KEYBRD= 00C8        KEY1  = 0078    KEY10 = 006D      KEY11 = 006E
KEY12 = 006F        KEY13 = 005C    KEY14 = 005D      KEY15 = 005E
KEY16 = 005F        KEY17 = 003C    KEY18 = 003D      KEY19 = 003E
KEY2  = 0079        KEY20 = 003F    KEY3  = 007A      KEY4  = 007B
KEY5  = 0074        KEY6  = 0075    KEY7  = 0076      KEY8  = 0077
KEY9  = 006C        KINTA = 006F    KLDAD = 003D      KN0   = 003C
KN1   = 005C        KN2   = 005D    KN3   = 005E      KN4   = 006C
KN5   = 006D        KN6   = 006E    KN7   = 0074      KN8   = 0075
KN9   = 0076        KU2   = 0079    KU3   = 007A      KU8   = 0077
L     =%0005        LBLANK= 000F    LCE   = 000B      LCH   = 000C
LCL   = 000D        LCP   = 000E    LC0   = 0000      LC1   = 0001
LC2   = 0002        LC3   = 0003    LC4   = 0004      LC5   = 0005
LC6   = 0006        LC7   = 0007    LC8   = 0008      LC9   = 0009
LDLEDA= 00CA        LDLEDB= 00CB    LDLEDC= 00CC      LDLEDD= 00CD
LDLEDE= 00CE        LDLEDF= 00CF    LKDIAG= 2800      LKKBD = 004C
LKKEY = 0049        LKLWMG= 0058    LKLWMP= 0055      LKLWPG= 0052
LKLWPP= 004F        LKMOD7= 0046    LKOPR = 0046      LPFLG   4F94
LPNUM   4F92        M     =%0006    MBSEL = 00E0      MB.A  = 0008
MB.B  = 0004        MEMTOP= 4FFF    MINUS = 000A      MM    = 8000
MSE   = 0008        MSGN  = 001D    MTACLR= 0000      MT.ARA= 0020
MT.CPE= 0080        MT.DSE= 0001    MT.FWD= 0040      MT.INH= 0008
MT.LWR= 0004        MT.MOT= 0002    MT.NWT= 0080      MT.PEC= 0040
MT.PSB= 0004        MT.PS0= 0001    MT.PS1= 0002      MT.REV= 0020
MT.WRT= 0010        MT.Z  = 0008    M.ATA = 0080      M.CAPE= 0020
M.CONT= 0080        M.DEM = 0020    M.EBL = 0010      M.EXC = 0008
M.FAIL= 0008        M.ILR = 0010    M.INIT= 0010      M.OCC = 0020
M.ONLI= 0001        M.PE  = 0040    M.PORT= 0080      M.RDEN= 0002
M.RDPE= 0008        M.RUN = 0004    M.SCLK= 0001      M.TRA = 0040
M.UNIT= 0007        M.WCLK= 0040    M.WCLN= 0080      M.WREN= 0080
M5.5  = 0001        M6.5  = 0002    M7.5  = 0004      NOTCAP= 0088
OKAY  = 00FF        OPRRAM= 4300    OPSTRT= 0058      OPVER = 0040
PADCNT= 00D5        PADCRC= 0080    PDIAG = 0048      PEID  = 008A
PENAB = 004C        PESET = 0001    PL    = 00B1      PRDD  = 004C
PRENF = 009C        PS    = 00B2    PSTAT = 0048      PSW   =%0009
P.AMTP= 0001        P.BCTC= 0040    P.CMDP= 0020      P.INTE= 0080
P.LCS = 0040        P.LWR = 0020    P.RDP = 0002      P.RPEN= 0004
P.RPST= 0002        P.RP0E= 0020    P.RP1E= 0010      P.RP2E= 0020
P.RP3E= 0010        P.SING= 0080    P.STAT= 0002      P.STPE= 0080
P.TACH= 0008        P.TUPR= 0010    P.WCSP= 0004      P.WDS = 0040
P.WFLP= 0001        P.WPEN= 0010    P.WP0E= 0008      P.WP1E= 0004
P.WP2E= 0008        P.WP3E= 0004    P.5VOK= 0002      QUE   = 281B
QUEM  = 281E        RAMT  = 0010    RARA  = 0006      RARAI = 0004
RCHBD0= 0048        RCHBD1= 0047    RCHOK = 0046      RCHTST= 000C
RCLRT = 000D        RCMD  = 0008    RCMLP = 0003      RCONT = 0080
RDATA = 0017        RDCLK = 0010    RDON  = 0011      READG = 0007
REND  = 0014        REQST = 2806    RESCHR= 00D1      REVTST= 0064
REWIND= 0004        RFIFOL= 0008    RGCLK = 0002      RGCRI = 0003
RIBG  = 0001        RILL  = 0012    RINST = 000C      RMCTST= 0008
RMK2  = 0013        RNOP  = 0000    RPATH = 0001      RPBAD = 0044
RPCHI = 0001        RPCLK = 0003    RPCTL = 0009      RPEI  = 0002
```

MTA1 - MAG TAPE ADAPTER TEST PART #1 CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:02 PAGE 1-42 SEQ 0330
MTA1.M80 SYMBOL TABLE

```
RPFAIL= 0000        RPF1  = 009D        RPF2  - 009E        RPOK  = 0043
RPOSTN= 0016        RPSTA = 0015        RRCMT = 000A        RSTAT = 0002
RTIEB = 000A        RTIER = 0030        RTM   = 0005        RUNKI = 0009
RUP1ST= 005E        RWDUNL= 0005        R.AMT = 0001        R.BOP = 0008
R.DATA= 0040        R.DON = 0002        R.DRDY= 0010        R.END = 0010
R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008        R.PLOD= 0008
R.PL00= 0010        R.PL01= 0020        R.POST= 0020        R.STNM= 0002
R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040
R.VOK = 0080        R00H  = 0081        R00L  = 0080        R01H  = 0083
R01L  = 0082        R02H  = 0085        R02L  = 0084        R03H  = 0087
R03L  = 0086        R04H  = 0089        R04L  = 0088        R05H  = 0088
R05L  = 008A        R06H  = 008D        R06L  = 008C        R07H  = 008F
R07L  = 008E        R10H  = 0091        R10L  = 0090        R11H  = 0093
R11L  = 0092        R12H  = 0095        R12L  = 0094        R13H  = 0097
R13L  = 0096        R14H  = 0099        R14L  = 0098        R15H  = 0098
R15L  = 009A        R16H  = 009D        R16L  = 009C        R17H  = 009F
R17L  = 009E        R7.5  = 0010        SELCLR= 0000        SETATA= 00A1
SID   = 0080        SOD   = 0080        SOE   = 0040        SP    =%0008
SSCLK = 0040        SSTEP = 0005        STACK = 4FFF        STATRM= 4F20
STPCT   4F20        STRSP = 5000        TADR00= 0080        TADR01= 0081
TADR02= 0082        TADR03= 0083        TADR04= 0084        TADR05= 0085
TADR06= 0086        TADR07= 0087        TADR10= 0088        TADR11= 0089
TADR12= 008A        TADR13= 008B        TAMT  = 0044        TASEL = 0080
TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004
TC.REV= 0020        TC.WRT= 0010        TEMP    4F99        TEST01  4300
TEST02  43A5        TEST03  442A        TEST04  44AF        TEST05  4534
TEST06  45BB        TEST07  4644        TEST10  46CD        TEST11  4756
TEST12  47E6        TEST13  4878        TEST14  490A        TMF   = 0099
TMRDY = 0040        TRKENA= 00D2        TSET  = 2803        TSTEND= 2818
TSTS  = 0040        TST01C  4344  G     TST01X  4327        TST02C  43C9  G
TST02X  43AA        TST03C  444E  G     TST03X  442F        TST04C  44D3  G
TST04X  44B4        TST05X  4539        TST06X  45C0        TST07X  4649
TST10X  46D2        TST11X  475B        TST12X  47EB        TST13X  487D
TST14X  490F        TUSEL0= 00D1        TUSEL1= 00D2        TU78  = 0010
T.ATH0= 0001        T.ATH1= 0002        T.BOT = 0004        T.EOT = 0002
T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020        T.PES = 0008
T.PSBJ= 0020        T.PSOJ= 0008        T.PS1J= 0010        T.RDY = 0080
T.RDY0= 0040        T.RWD = 0010        T.SCLK= 0002        UIBG  = 00A1
UNITMP  49CD        VALFC   4F98        VALTB   4F95        VELTST= 005B
WDR.P = 0010        WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0
WRTCLK= 0000        WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008
W.DIAG= 0002        W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080
W.ERR = 0020        W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004
W.ONES= 0020        W.RESI= 0002        W.REV = 0004        W.ROME= 0010
W.RST = 0001        W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020
X     =%000A        X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002
X.ROME- 0001        X.WCLK= 0001        Y     =%000B        .     = 49CF


ERRORS DETECTED:  0

*MTA1.A78/PTP,MTA1=NLIST,PARAM,MACRO,LIST,MTA1
RUN-TIME: 5 7 0 SECONDS
```

CORE USED: 10K

TABLE OF CONTENTS

```
                                .TITLE  MTA2 - MAG TAPE ADAPTER TEST PART #2
                                ;ID MTA2-MAG TAPE ADAPTER CONTROLLER PART #2
                                .SBTTL  TEST 01 - MASSBUS PORT SELECT SWITCH TEST - TU #0
4300                            ST
 (1)                            ; **********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*MASSBUS PORT SELECT SWITCH TEST - TU #0
4300                            SD
 (1)                            ; **********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SE ECT SWITCH AND
                                ;*ASSOCIATED HARDWARE, FOR TU #0.
4300                            SP
 (1)                            ; **********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*   IF TU PORT #0 NOT SELECTED BY USER
                                ;*   :  THEN-NEXT TEST
                                ;*   :  ELSE-CONTINUE
                                ;*   ENDIF
                                ;*   IF MANUAL INTERVENTION NOT ALLOWED
                                ;*   :  THEN-NEXT TEST
                                ;*   :  ELSE-CONTINUE
                                ;*   ENDIF
                                ;*   CALL SUBROUTINE CLEAR
                                ;*   SELECT TU PORT #0
                                ;*   SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #0
                                ;*   REQUEST THE USER TO SET TU78 #0 TO MASSBUS PORT 0
                                ;*   INPUT MTA #0 STATUS REGISTER B
                                ;*   IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-ERROR
                                ;*   ENDIF
                                ;*   REQUEST THE USER TO SET TU78 #0 TO MASSBUS PORT 1
                                ;*   INPUT MTA #0 STATUS REGISTER B
                                ;*   IF PSB NOT AND PS0 NOT=1 AND PS1 NOT=0
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-ERROR
                                ;*   ENDIF
                                ;*   REQUEST THE USER TO SET TU78 #0 TO BOTH MASSBUS PORTS
                                ;*   INPUT MTA #0 STATUS REGISTER B
                                ;*   IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT-1
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-ERROR
                                ;*   ENDIF
                                ;*   REQUEST THE USER TO SET TU78 #0 TO MAINTENANCE MODE
                                ;*   INPUT MTA #0 STATUS REGISTER B
                                ;*   IF PSB NOT PS0 NOT AND PS1 NOT=1
                                ;*   :  THEN-CONTINUE
                                ;*   :  ELSE-ERROR
                                ;*   ENDIF
```

```
                                        ;*ENDTST
4300                                    SE
                                        ; ***************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 01
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 03
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
                                        ;*ACUTAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 05
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 07
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 11
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
                                        ; ***************************************************************************

                                        TEST01: TESTX   @01             ;INITIALIZE THE TEST
4300  3E  01              7.0                       MVI     A,@01         ;DEFINE THE TEST NUMBER
4302  CD  03  28          18.0                      CALL    TSET          ;SETUP THE TEST
                                        ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;&M8954, TU CABLE, M8955'S
4305                                            REQ     @7,0,0,0,0
4305  CD  06  28          18.0                      CALL    REQST
4308  00                                          .BYTE   0             ;DATA PATTERN NUMBER
```

```
 (1)  4309  00  00                                        .WORD   0               ;SYSTEM '0' COUNT
 (1)  430B  00  00                                        .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                            .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                            .BYTE   @7              ;REQUEST CODE
1420  430F                                    RIN     R12L                    ;GET THE UNITS DESIRED
 (1)  430F  DB  94              10.0                     IN      R12L                    ;READ R12L INTO AC
 (1)  4311  7F                   4.0                     MOV     A,A                     ;RETRY LINK
1421  4312  32  0A  4A           13.0            STA     UNITMP                  ;STORE IN MEMORY
1422  4315  A7                    4.0            ANA     A                       ;SET THE CONDITION CODF
1423  4316  C2  27  43           10.0            JNZ     TST01X                  ;GO TEST THE SPECIFIED UNITS
1424                                                                            ;NO UNITS SPECIFIED
1425  4319                                    ERR     EXIT,DUMMY
 (1)                                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28          18.0                     CALL    ERLP                    ;PROCESS ERROR - DO 2.3
 (1)        0001                                         MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                           .BYTE
 (1)  431E  CD  15  28          18.0                DUMMY:: CALL    CKLOP                   ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  6D  48          10.0                    JC      EXIT                    ;LOOP ADDRESS IF LOOP SPECIFIED
1426                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1427                                        ;>FATAL ERROR - TEST ABORTED
1428  4324  C3  6D  48          10.0            JMP     EXIT                    ;EXIT THE TEST
1429
1430  4327  3A  0A  4A           13.0    TST01X: LDA     UNITMP                  ;GET THE UNIT MAP
1431  432A  E6  01                 7.0            ANI     @01                     ;TEST UNIT 0
1432  432C  CA  E6  43           10.0            JZ      TEST02                  ;NO-CHECK FOR TEST 02
1433  432F  CD  70  48           18.0            CALL    CLEAR                   ;CLEAR THE TU PORTS
1434  4332  DB  E0                10.0            IN      INTSTA                  ;SELECT TU PORT 0
1435  4334  E6  80                 7.0            ANI     BIT7
1436  4336  F6  00                 7.0            ORI     @00
1437  4338  D3  E0                10.0            OUT     MBSEL
1438  433A  3E  83                 7.0            MVI     A,$83                   ;SELECT MTA 0 REGISTER 3
1439  433C  D3  40                10.0            OUT     TCMD
1440  433E  3E  38                 7.0            MVI     A,$38                   ;SET THE PS0 NOT, PS1 NOT,
1441  4340  D3  40                10.0            OUT     TCMD                    ;AND PSB NOT BITS.
1442  4342                                    QMSGM   TEST02
 (1)  4342  CD  1E  28          18.0            CALL    QUEM
 (1)        0002                                MSGN=MSGN+1
 (1)  4345  02                                  .BYTE   MSGN
 (1)  4346  DA  E6  43          10.0            JC      TEST02
1443                                        ;>SET THE TU78 #0 PORT SELECT SWITCH TO MASSBUS PORT 0
1444  4349  3E  82                 7.0            MVI     A,$82                   ;SELECT MTA 0 REGISTER 2
1445  434B  D3  40                10.0            OUT     TCMD
1446  434D  DB  40                10.0            IN      TSTS                    ;GET THE TU PORT STATUS
1447  434F  E6  07                 7.0            ANI     $07                     ;SAVE ONLY PORT SELECT BITS
1448  4351  FE  06                 7.0            CPI     BIT1!BIT2               ;COMPARE WITH EXPECTED
1449  4353  CA  63  43           10.0            JZ      T1CN1                   ;BRANCH IF EQUAL
1450  4356                                        ROUT    ADATA                   ;SAVE THE ACTUAL DATA
 (1)  4356  D3  94                10.0                    OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  4358  7F                    4.0                    MOV     A,A                     ;RETRY LINK
1451  4359  3E  06                 7.0            MVI     A,BIT1!BIT2             ;SAVE THE EXPECTED DATA
1452  435B                                        ROUT    EDATA
 (1)  435B  D3  95                10.0                    OUT     EDATA                   ;WRITE AC INTO EDATA
 (1)  435D  7F                    4.0                    MOV     A,A                     ;RETRY LINK
```

```
1453  435E                                          ERRB    TST01X,T1CN1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  435E  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0003                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4361  03                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4362  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4363  CD  15  28        18.0                  T1CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4366  DA  27  43        10.0                          JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
1454                                        ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1455
1456  4369                                          QMSGM   TEST02
 (1)  4369  CD  1E  28        18.0                  CALL    QUEM
 (1)        0004                                    MSGN=MSGN+1
 (1)  436C  04                                      .BYTE   MSGN
 (1)  436D  DA  E6  43        10.0                  JC      TEST02
1457                                        ;>SET THE TU78 #0 PORT SELECT SWITCH TO MASSBUS PORT 1
1458  4370  DB  40              10.0                  IN      TSTS
1459  4372  E6  07               7.0                  ANI     $07
1460  4374  FE  05               7.0                  CPI     BIT0!BIT2
1461  4376  CA  86  43          10.0                  JZ      T1CN2
1462  4379                                          ROUT    ADATA
 (1)  4379  D3  94              10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  437B  7F                    4.0                         MOV     A,A             ;RETRY LINK
1463  437C  3E  05               7.0                  MVI     A,BIT0!BIT2
1464  437E                                          ROUT    EDATA
 (1)  437E  D3  95              10.0                          OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4380  7F                    4.0                         MOV     A,A             ;RETRY LINK
1465  4381                                          ERRB    TST01X,T1CN2
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4381  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4384  05                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4385  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4386  CD  15  28        18.0                  T1CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4389  DA  27  43        10.0                          JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
1466                                        ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
1467
1468  438C                                          QMSGM   TEST02
 (1)  438C  CD  1E  28        18.0                  CALL    QUEM
 (1)        0006                                    MSGN=MSGN+1
 (1)  438F  06                                      .BYTE   MSGN
 (1)  4390  DA  E6  43        10.0                  JC      TEST02
1469                                        ;>SET THE TU78 #0 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
1470  4393  DB  40              10.0                  IN      TSTS
1471  4395  E6  07               7.0                  ANI     $07
1472  4397  FE  03               7.0                  CPI     BIT0!BIT1
1473  4399  CA  A9  43          10.0                  JZ      T1CN3
1474  439C                                          ROUT    ADATA
 (1)  439C  D3  94              10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  439E  7F                    4.0                         MOV     A,A             ;RETRY LINK
1475  439F  3E  03               7.0                  MVI     A,BIT0!BIT1
1476  43A1                                          ROUT    EDATA
 (1)  43A1  D3  95              10.0                          OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  43A3  7F                    4.0                         MOV     A,A             ;RETRY LINK
```

```
                                        ERRB    TST01X,T1CN3
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1)  43A4  CD  12  28      18.0                         .BYTE   MSGN            ;MESSAGE NUMBER ID
(1)        0007
(1)  43A7  07                                           .BYTE                   ;PRINT ROUTINE NUMBER
(1)  43A8  00
(1)  43A9  CD  15  28      18.0                 T1CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1)  43AC  DA  27  43      10.0                         JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
                                        QMSGM   TEST02
(1)  43AF  CD  1E  28      18.0         CALL    QUEM
(1)        0008                         MSGN=MSGN+1
(1)  43B2  08                           .BYTE   MSGN
(1)  43B3  DA  E6  43      10.0         JC      TEST02
                                ;>SET THE TU78 #0 PORT SELECT SWITCH TO MAINTENANCE
     43B6  DB  40          10.0         IN      TSTS
     43B8  E6  07           7.0         ANI     $07
     43BA  FE  07           7.0         CPI     BIT0!BIT1!BIT2
     43BC  CA  CC  43      10.0         JZ      T1CN4
     43BF                               ROUT    ADATA
(1)  43BF  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
(1)  43C1  7F               4.0                 MOV     A,A             ;RETRY LINK
     43C2  3E  07           7.0         MVI     A,BIT0!BIT1!BIT2
     43C4                               ROUT    EDATA
(1)  43C4  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
(1)  43C6  7F               4.0                 MOV     A,A             ;RETRY LINK
     43C7                               ERRB    TST01X,T1CN4
(1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
(1)  43C7  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
(1)        0009                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1)  43CA  09                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
(1)  43CB  00                                   .BYTE                   ;PRINT ROUTINE NUMBER
(1)  43CC  CD  15  28      18.0         T1CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1)  43CF  DA  27  43      10.0                 JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
                                        ENDTST  TST01X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
(1)                                     REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
(2)  43D2                                       CALL    REQST
(2)  43D2  CD  06  28      18.0                 .BYTE                   ;DATA PATTERN NUMBER
(2)  43D5  00                                   .WORD                   ;SYSTEM '0' COUNT
(2)  43D6  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2)  43D8  00  00                               .BYTE                   ;DATA COMPARE FLAG IF =1
(2)  43DA  00                                   .BYTE   7               ;REQUEST CODE
(2)  43DB  07
(1)  43DC  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
(1)  43DF  3D               4.0         DCR     A               ;DOWNCOUNT
(1)  43E0  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
(1)  43E3  F2  27  43      10.0         JP      TST01X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 02 - MASSBUS PORT SELECT SWITCH TEST - TU #1
43E6                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*MASSBUS PORT SELECT SWITCH TEST - TU #1
43E6                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SELECT SWITCH AND
                                ;*ASSOCIATED HARDWARE, FOR TU #1.
43E6                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #1 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  IF MANUAL INTERVENTION NOT ALLOWED
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #1
                                ;*  SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #1
                                ;*  REQUEST THE USER TO SET TU78 #1 TO MASSBUS PORT 0
                                ;*  INPUT MTA #1 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #1 TO MASSBUS PORT 1
                                ;*  INPUT MTA #1 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS0 NOT=1 AND PS1 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #1 TO BOTH MASSBUS PORTS
                                ;*  INPUT MTA #1 STATUS REGISTER B
                                ;*  IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #1 TO MAINTENANCE MODE
                                ;*  INPUT MTS #1 STATUS REGISTER B
                                ;*  IF PSB NOT PS0 NOT AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDTST
43E6                            SE
```

```
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1538                                ;*MTA2 MICRO TEST 02
1539                                ;*MTA2 MICRO ERROR 13
1540                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1541                                ;*M8954, TU CABLE, M8955'S
1542                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1543                                ;*ACTUAL = NNNN
1544                                ;*EXPECTED = NNNN
1545                                ;*
1546                                ;*MTA2 MICRO TEST 02
1547                                ;*MTA2 MICRO ERROR 15
1548                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1549                                ;*M8954, TU CABLE, M8955'S
1550                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
1551                                ;*ACTUAL = NNNN
1552                                ;*EXPECTED = NNNN
1553                                ;*
1554                                ;*MTA2 MICRO TEST 02
1555                                ;*MTA2 MICRO ERROR 17
1556                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1557                                ;*M8954, TU CABLE, M8955'S
1558                                ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1559                                ;*ACTUAL = NNNN
1560                                ;*EXPECTED = NNNN
1561                                ;*
1562                                ;*MTA2 MICRO TEST 02
1563                                ;*MTA2 MICRO ERROR 21
1564                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1565                                ;*M8954, TU CABLE, M8955'S
1566                                ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
1567                                ;*ACTUAL = NNNN
1568                                ;*EXPECTED = NNNN
1569  43E6                          S
 (1)                                ; ***************************************************************************
1570
1571  43E6                          TEST02: TESTX   @02             ;INITIALIZE THE TEST
 (1)  43E6  3E  02          7.0                  MVI     A,@02           ;DEFINE THE TEST NUMBER
 (1)  43E8  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
1572                                ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1573                                ;&M8954, TU CABLE, M8955'S
1574  43EB  3A  0A  4A     13.0    TST02X: LDA     UNITMP          ;GET THE UNIT MAP
1575  43EE  E6  02          7.0            ANI     @02             ;TEST UNIT 1
1576  43F0  CA  AA  44      10.0            JZ      TEST03          ;NO-CHECK FOR TEST 03
1577  43F3  CD  70  48      18.0            CALL    CLEAR           ;CLEAR THE TU PORTS
1578  43F6  DB  E0          10.0            IN      INTSTA          ;SELECT TU PORT 1
1579  43F8  E6  80           7.0            ANI     BIT7
1580  43FA  F6  01           7.0            ORI     @01
1581  43FC  D3  E0          10.0            OUT     MBSEL
1582  43FE  3E  83           7.0            MVI     A,$83           ;SELECT MTA 1 REGISTER 3
1583  4400  D3  40          10.0            OUT     TCMD
1584  4402  3E  38           7.0            MVI     A,$38           ;SET THE PS0 NOT, PS1 NOT,
1585  4404  D3  40          10.0            OUT     TCMD            ;AND PSB NOT BITS.
```

```
1586  4406                                          QMSGM   TEST03
 (1)  4406  CD  1E  28        18.0                CALL    QUEM
 (1)        000A                                  MSGN=MSGN+1
 (1)  4409  0A                                    .BYTE   MSGN
 (1)  440A  DA  AA  44        10.0                JC      TEST03
1587                                        ;>SET THE TU78 #1 PORT SELECT SWITCH TO MASSBUS PORT 0
1588  440D  3E  82                7.0           MVI     A,$82           ;SELECT MTA 1 REGISTER 2
1589  440F  D3  40               10.0           OUT     TCMD
1590  4411  DB  40               10.0           IN      TSTS            ;GET THE TU PORT STATUS
1591  4413  E6  07                7.0           ANI     $07             ;SAVE ONLY PORT SELECT BITS
1592  4415  FE  06                7.0           CPI     BIT1!BIT2       ;COMPARE WITH EXPECTED
1593  4417  CA  27  44           10.0           JZ      T2CN1           ;BRANCH IF EQUAL
1594  441A                                      ROUT    ADATA           ;SAVE THE ACTUAL DATA
 (1)  441A  D3  94               10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  441C  7F                    4.0                   MOV     A,A             ;RETRY LINK
1595  441D  3E  06                7.0           MVI     A,BIT1!BIT2     ;SAVE THE EXPECTED DATA
1596  441F                                      ROUT    EDATA
 (1)  441F  D3  95               10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4421  7F                    4.0                   MOV     A,A             ;RETRY LINK
1597  4422                                      ERRB    TST02X,T2CN1
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4422  CD  12  28        18.0                           CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000B                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4425  0B                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4426  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4427  CD  15  28        18.0                 T2CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  442A  DA  EB  43        10.0                         JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1598                                        ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1599  442D                                          QMSGM   TEST03
 (1)  442D  CD  1E  28        18.0                CALL    QUEM
 (1)        000C                                  MSGN=MSGN+1
 (1)  4430  0C                                    .BYTE   MSGN
 (1)  4431  DA  AA  44        10.0                JC      TEST03
1600                                        ;>SET THE TU78 #1 PORT SELECT SWITCH TO MASSBUS PORT 1
1601  4434  DB  40               10.0           IN      TSTS
1602  4436  E6  07                7.0           ANI     $07
1603  4438  FE  05                7.0           CPI     BIT0!BIT2
1604  443A  CA  4A  44           10.0           JZ      T2CN2
1605  443D                                      ROUT    ADATA
 (1)  443D  D3  94               10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  443F  7F                    4.0                   MOV     A,A             ;RETRY LINK
1606  4440  3E  05                7.0           MVI     A,BIT0!BIT2
1607  4442                                      ROUT    EDATA
 (1)  4442  D3  95               10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4444  7F                    4.0                   MOV     A,A             ;RETRY LINK
1608  4445                                      ERRB    TST02X,T2CN2
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4445  CD  12  28        18.0                           CALL    ERLPR           ;PROCESS ERROR - DO 2.3
 (1)        000D                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4448  0D                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4449  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  444A  CD  15  28        18.0                 T2CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  444D  DA  EB  43        10.0                         JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1609                                        ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
```

```
1610
1611  4450                                          QMSGM   TEST03
 (1)  4450  CD  1E  28        18.0                 CALL    QUEM
 (1)        000E                                   MSGN=MSGN+1
 (1)  4453  0E                                     .BYTE   MSGN
 (1)  4454  DA  AA  44        10.0                 JC      TEST03
1612                                    ;>SET THE TU78 #1 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
1613  4457  DB  40              10.0                IN      TSTS
1614  4459  E6  07               7.0                ANI     $07
1615  445B  FE  03               7.0                CPI     BIT0!BIT1
1616  445D  CA  6D  44           10.0                JZ      T2CN3
1617  4460                                          ROUT    ADATA
 (1)  4460  D3  94              10.0                        OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4462  7F                   4.0                        MOV     A,A             ;RETRY LINK
1618  4463  3E  03               7.0                MVI     A,BIT0!BIT1
1619  4465                                          ROUT    EDATA
 (1)  4465  D3  95              10.0                        OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4467  7F                   4.0                        MOV     A,A             ;RETRY LINK
1620  4468                                          ERRB    TST02X,T2CN3
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4468  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  446B  0F                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  446C  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  446D  CD  15  28        18.0                T2CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4470  DA  EB  43        10.0                        JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1621                                    ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1622
1623  4473                                          QMSGM   TEST03
 (1)  4473  CD  1E  28        18.0                 CALL    QUEM
 (1)        0010                                   MSGN=M GN+1
 (1)  4476  10                                     .BYTE   MSGN
 (1)  4477  DA  AA  44        10.0                 JC      TEST03
1624                                    ;>SET THE TU78 #1 PORT SELECT SWITCH TO MAINTENANCE
1625  447A  DB  40              10.0                IN      TSTS
1626  447C  E6  07               7.0                ANI     $07
1627  447E  FE  07               7.0                CPI     BIT0!BIT1!BIT2
1628  4480  CA  90  44           10.0                JZ      T2CN4
1629  4483                                          ROUT    ADATA
 (1)  4483  D3  94              10.0                        OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4485  7F                   4.0                        MOV     A,A             ;RETRY LINK
1630  4486  3E  07               7.0                MVI     A,BIT0!BIT1!BIT2
1631  4488                                          ROUT    EDATA
 (1)  4488  D3  95              10.0                        OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  448A  7F                   4.0                        MOV     A,A             ;RETRY LINK
1632  448B                                          ERRB    TST02X,T2CN4
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  448B  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0011                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448E  11                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  448F  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4490  CD  15  28        18.0                T2CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4493  DA  EB  43        10.0                        JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1633                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
```

```
                                        ENDTST  TST02X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
1634  4496                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (1)
 (2)  4496                                              CALL    REQST
 (2)  4496  CD  06  28      18.0                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  4499  00                                          .WORD                   ;SYSTEM "" COUNT
 (2)  449A  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  449C  00  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  449E  00                                          .BYTE   7               ;REQUEST CODE
 (2)  449F  07
 (1)  44A0  3A  9A  4F      13.0        LDA     ITERA       ;GET ITERATION COUNT
 (1)  44A3  3D               4.0        DCR     A           ;DOWNCOUNT
 (1)  44A4  32  9A  4F      13.0        STA     ITERA       ;SAVE COUNT
 (1)  44A7  F2  EB  43      10.0        JP      TST02X      ;DO TEST UNTIL TILL - 0
```

```
1636                            .SBTTL  TEST 03 - MASSBUS PORT SELECT SWITCH TEST - TU #2
1637  44AA                      ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1638                            ;*MASSBUS PORT SELECT SWITCH TEST - TU #2
1639  44AA                      SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1640                            ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SELECT SWITCH AND
1641                            ;*ASSOCIATED HARDWARE, FOR TU #2.
1642  44AA                      SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1643                            ;*BGNTST
1644                            ;*   IF TU PORT #2 NOT SELECTED BY USER
1645                            ;*   :  THEN-NEXT TEST
1646                            ;*   :  ELSE-CONTINUE
1647                            ;*   ENDIF
1648                            ;*   IF MANUAL INTERVENTION NOT ALLOWED
1649                            ;*   :  THEN-NEXT TEST
1650                            ;*   :  ELSE-CONTINUE
1651                            ;*   ENDIF
1652                            ;*   CALL SUBROUTINE CLEAR
1653                            ;*   SELECT TU PORT #2
1654                            ;*   SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #2
1655                            ;*   REQUEST THE USER TO SET TU78 #2 TO MASSBUS PORT 0
1656                            ;*   INPUT MTA #2 STATUS REGISTER B
1657                            ;*   IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
1658                            ;*   :  THEN-CONTINUE
1659                            ;*   :  ELSE-ERROR
1660                            ;*   ENDIF
1661                            ;*   REQUEST THE USER TO SET TU78 #2 TO MASSBUS PORT 1
1662                            ;*   INPUT MTA #2 STATUS REGISTER B
1663                            ;*   IF PSB NOT AND PS0 NOT=1 AND PS1 NOT=0
1664                            ;*   :  THEN-CONTINUE
1665                            ;*   :  ELSE-ERROR
1666                            ;*   ENDIF
1667                            ;*   REQUEST THE USER TO SET TU78 #2 TO BOTH MASSBUS PORTS
1668                            ;*   INPUT MTA #2 STATUS REGISTER B
1669                            ;*   IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT=1
1670                            ;*   :  THEN-CONTINUE
1671                            ;*   :  ELSE-ERROR
1672                            ;*   ENDIF
1673                            ;*   REQUEST THE USER TO SET TU78 #2 TO MAINTENANCE MODE
1674                            ;*   INPUT MTS #2 STATUS REGISTER B
1675                            ;*   IF PSB NOT PS0 NOT AND PS1 NOT=1
1676                            ;*   :  THEN-CONTINUE
1677                            ;*   :  ELSE-ERROR
1678                            ;*   ENDIF
1679                            ;*ENDTST
1680  44AA                      SE
```

```
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1681                                ;*MTA2 MICRO TEST 03
1682                                ;*MTA2 MICRO ERROR 23
1683                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1684                                ;*M8954, TU CABLE, M8955'S
1685                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1686                                ;*ACTUAL = NNNN
1687                                ;*EXPECTED = NNNN
1688                                ;*
1689                                ;*MTA2 MICRO TEST 03
1690                                ;*MTA2 MICRO ERROR 25
1691                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1692                                ;*M8954, TU CABLE, M8955'S
1693                                ;*PORT SELECT DOES NOT INDICATE MASSBUS PORT 1
1694                                ;*ACTUAL = NNNN
1695                                ;*EXPECTED = NNNN
1696                                ;*
1697                                ;*MTA2 MICRO TEST 03
1698                                ;*MTA2 MICRO ERROR 27
1699                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1700                                ;*M8954, TU CABLE, M8955'S
1701                                ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1702                                ;*ACTUAL = NNNN
1703                                ;*EXPECTED = NNNN
1704                                ;*
1705                                ;*MTA2 MICRO TEST 03
1706                                ;*MTA2 MICRO ERROR 31
1707                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1708                                ;*M8954, TU CABLE, M8955'S
1709                                ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
1710                                ;*ACTUAL = NNNN
1711                                ;*EXPECTED = NNNN
1712  44AA                          S
 (1)                                ; ***************************************************************************
1713
1714  44AA                          TEST03: TESTX   @03             ;INITIALIZE THE TEST
 (1)  44AA  3E  03          7.0                     MVI     A,@03           ;DEFINE THE TEST NUMBER
 (1)  44AC  CD  03  28     18.0                     CALL    TSET            ;SETUP THE TEST
1715                                ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1716                                ;&M8954, TU CABLE, M8955'S
1717  44AF  3A  0A  4A     13.0    TST03X: LDA     UNITMP          ;GET THE UNIT MAP
1718  44B2  E6  04          7.0            ANI     @04             ;TEST UNIT 2
1719  44B4  CA  6E  45      10.0            JZ      TEST04          ;NO-CHECK FOR TEST 04
1720  44B7  CD  70  48      18.0            CALL    CLEAR           ;CLEAR THE TU PORTS
1721  44BA  DB  E0          10.0            IN      INTSTA          ;SELECT TU PORT 2
1722  44BC  E6  80           7.0            ANI     BIT7
1723  44BE  F6  02           7.0            ORI     @02
1724  44C0  D3  E0          10.0            OUT     MBSEL
1725  44C2  3E  83           7.0            MVI     A,$83           ;SELECT MTA 2 REGISTER 3
1726  44C4  D3  40          10.0            OUT     TCMD
1727  44C6  3E  38           7.0            MVI     A,$38           ;SET THE PS0 NOT, PS1 NOT,
1728  44C8  D3  40          10.0            OUT     TCMD            ;AND PSB NOT BITS.
```

```
                                        QMSGM   TEST04
                                        CALL    QUEM
                                        MSGN=MSGN+1
                                        .BYTE   MSGN
                                        JC      TEST04
                                ;>SET THE TU78 #2 PORT SELECT SWITCH TO MASSBUS PORT 0
                                        MVI     A,$82           ;SELECT MTA 2 REGISTER 2
                                        OUT     TCMD
                                        IN      TSTS            ;GET THE TU PORT STATUS
                                        ANI     $07             ;SAVE ONLY PORT SELECT BITS
                                        CPI     BIT1!BIT2       ;COMPARE WITH EXPECTED
                                        JZ      T3CN1           ;BRANCH IF EQUAL
                                        ROUT    ADATA           ;SAVE THE ACTUAL DATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MVI     A,BIT1!BIT2     ;SAVE THE EXPECTED DATA
                                        ROUT    EDATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    TST03X,T3CN1
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                T3CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
                                        QMSGM   TEST04
                                        CALL    QUEM
                                        MSGN=MSGN+1
                                        .BYTE   MSGN
                                        JC      TEST04
                                ;>SET THE TU78 #2 PORT SELECT SWITCH TO MASSBUS PORT 1
                                        IN      TSTS
                                        ANI     $07
                                        CPI     BIT0!BIT2
                                        JZ      T3CN2
                                        ROUT    ADATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MVI     A,BIT0!BIT2
                                        ROUT    EDATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    TST03X,T3CN2
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                T3CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
```

```
                                        QMSGM   TEST04
 4514                                   CALL    QUEM
 (1)  4514  CD  1E  28        18.0      MSGN=MSGN+1
 (1)        0016                        .BYTE   MSGN
 (1)  4517  16                          JC      TEST04
 (1)  4518  DA  6E  45        10.0  ;>SET THE TU78 #2 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
 1754
 1755  451B  DB  40           10.0      IN      TSTS
 1756  451D  E6  07            7.0     ANI     $07
 1757  451F  FE  03            7.0     CPI     BIT0!BIT1
 1758  4521  CA  31  45       10.0     JZ      T3CN3
 1759  4524                             ROUT    ADATA
 (1)   4524  D3  94           10.0              OUT     ADATA           ;WRITE AC INTO ADATA
 (1)   4526  7F                4.0              MOV     A,A             ;RETRY LINK
 1760  4527  3E  03            7.0     MVI     A,BIT0!BIT1
 1761  4529                             ROUT    EDATA
 (1)   4529  D3  95           10.0              OUT     EDATA           ;WRITE AC INTO EDATA
 (1)   452B  7F                4.0              MOV     A,A             ;RETRY LINK
 1762  452C                             ERRB    TST03X,T3CN3
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)   452C  CD  12  28       18.0              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0017                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)   452F  17                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)   4530  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)   4531  CD  15  28       18.0      T3CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)   4534  DA  AF  44       10.0              JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
 1763                           ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
 1764  4537                             QMSGM   TEST04
 (1)   4537  CD  1E  28       18.0      CALL    QUEM
 (1)        0018                        MSGN=MSGN+1
 (1)   453A  18                          .BYTE   MSGN
 (1)   453B  DA  6E  45       10.0      JC      TEST04
 1765                           ;>SET THE TU78 #2 PORT SELECT SWITCH TO MAINTENANCE
 1766  453E  DB  40           10.0      IN      TSTS
 1767  4540  E6  07            7.0     ANI     $07
 1768  4542  FE  07            7.0     CPI     BIT0!BIT1!BIT2
 1769  4544  CA  54  45       10.0     JZ      T3CN4
 1770  4547                             ROUT    ADATA
 (1)   4547  D3  94           10.0              OUT     ADATA           ;WRITE AC INTO ADATA
 (1)   4549  7F                4.0              MOV     A,A             ;RETRY LINK
 1771  454A  3E  07            7.0     MVI     A,BIT0!BIT1!BIT2
 1772  454C                             ROUT    EDATA
 (1)   454C  D3  95           10.0              OUT     EDATA           ;WRITE AC INTO EDATA
 (1)   454E  7F                4.0              MOV     A,A             ;RETRY LINK
 1773  454F                             ERRB    TST03X,T3CN4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)   454F  CD  12  28       18.0              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0019                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)   4552  19                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)   4553  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)   4554  CD  15  28       18.0      T3CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)   4557  DA  AF  44       10.0              JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
 1774                           ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
 1775  455A                       ENDTST  TST03X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
```

```
(2)  455A                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
(2)  455A  CD  06  28        18.0                       CALL    REQST
(2)  455D  00                                           .BYTE                   ;DATA PATTERN NUMBER
(2)  455E  00  00                                       .WORD                   ;SYSTEM '*' COUNT
(2)  4560  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2)  4562  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
(2)  4563  07                                           .BYTE   7               ;REQUEST CODE
(1)  4564  3A  9A  4F        13.0       LDA     ITERA           ;GET ITERATION COUNT
(1)  4567  3D                 4.0       DCR     A               ;DOWNCOUNT
(1)  4568  32  9A  4F        13.0       STA     ITERA           ;SAVE COUNT
(1)  4568  F2  AF  44        10.0       JP      TST03X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 04 - MASSBUS PORT SELECT SWITCH TEST - TU #3
1778  456E                      ST
                                ;*************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*MASSBUS PORT SELECT SWITCH TEST - TU #3
1780  456E                      SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SELECT SWITCH AND
                                ;*ASSOCIATED HARDWARE, FOR TU #3.
1783  456E                      SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #3 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  IF MANUAL INTERVENTION NOT ALLOWED
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #3
                                ;*  SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #3
                                ;*  REQUEST THE USER TO SET TU78 #3 TO MASSBUS PORT 0
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #3 TO MASSBUS PORT 1
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS0 NOT=1 AND PS1 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #3 TO BOTH MASSBUS PORTS
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #3 TO MAINTENANCE MODE
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT PS0 NOT AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDTST
1821  456E                      SE
```

```
(1)                                 ; ***************************************************************************
(1)                                 ;*ERRORS
(1)                                 ;*------
1822                                ;*MTA2 MICRO TEST 04
1823                                ;*MTA2 MICRO ERROR 33
1824                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1825                                ;*M8954, TU CABLE, M8955'S
1826                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1827                                ;*ACTUAL = NNNN
1828                                ;*EXPECTED = NNNN
1829                                ;*
1830                                ;*MTA2 MICRO TEST 04
1831                                ;*MTA2 MICRO ERROR 35
1832                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1833                                ;*M8954, TU CABLE, M8955'S
1834                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
1835                                ;*ACTUAL = NNNN
1836                                ;*EXPECTED = NNNN
1837                                ;*
1838                                ;*MTA2 MICRO TEST 04
1839                                ;*MTA2 MICRO ERROR 37
1840                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1841                                ;*M8954, TU CABLE, M8955'S
1842                                ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1843                                ;*ACTUAL = NNNN
1844                                ;*EXPECTED = NNNN
1845                                ;*
1846                                ;*MTA2 MICRO TEST 04
1847                                ;*MTA2 MICRO ERROR 4;
1848                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1849                                ;*M8954, TU CABLE, M8955'S
1850                                ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
1851                                ;*ACTUAL = NNNN
1852                                ;*EXPECTED = NNNN
1853  456E                          S
(1)                                 ; ***************************************************************************
1854
1855  456E                          TEST04: TESTX   @04             ;INITIALIZE THE TEST
(1)   456E  3E  04          7.0                 MVI     A,@04       ;DEFINE THE TEST NUMBER
(1)   4570  CD  03  28     18.0                 CALL    TSET        ;SETUP THE TEST
1856                                ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1857                                ;&M8954, TU CABLE, M8955'S
1858  4573  3A  0A  4A     13.0    TST04X: LDA     UNITMP          ;GET THE UNIT MAP
1859  4576  E6  08           7.0            ANI     @010            ;TEST UNIT 3
1860  4578  CA  32  46      10.0            JZ      TEST05          ;NO-EXIT TEST
1861  457B  CD  70  48      18.0            CALL    CLEAR           ;CLEAR THE TU PORTS
1862  457E  DB  E0          10.0            IN      INTSTA          ;SELECT TU PORT 3
1863  4580  E6  80           7.0            ANI     BIT7
1864  4582  F6  03           7.0            ORI     @03
1865  4584  D3  E0          10.0            OUT     MBSEL
1866  4586  3E  83           7.0            MVI     A,$83           ;SELECT MTA 3 REGISTER 3
1867  4588  D3  40          10.0            OUT     TCMD
1868  458A  3E  38           7.0            MVI     A,$38           ;SET THE PS0 NOT, PS1 NOT,
1869  458C  D3  40          10.0            OUT     TCMD            ;AND PSB NOT BITS.
```

```
                                                QMSGM   TEST05
 (1)  458E   CD   1E   28        18.0           CALL    QUEM
 (1)         001A                               MSGN=MSGN+1
 (1)  4591   1A                                 .BYTE   MSGN
 (1)  4592   DA   32   46        10.0           JC      TEST05
1871                                    ;>SET THE TU78 #3 PORT SELECT SWITCH TO MASSBUS PORT 0
1872  4595   3E   82                7.0          MVI     A,$82           ;SELECT MTA 3 REGISTER 2
1873  4597   D3   40               10.0          OUT     TCMD
1874  4599   DB   40               10.0          IN      TSTS            ;GET THE TU PORT STATUS
1875  459B   E6   07                7.0          ANI     $07             ;SAVE ONLY PORT SELECT BITS
1876  459D   FE   06                7.0          CPI     BIT1!BIT2       ;COMPARE WITH EXPECTED
1877  459F   CA   AF   45          10.0          JZ      T4CN1           ;BRANCH IF EQUAL
1878  45A2                                       ROUT    ADATA           ;SAVE THE ACTUAL DATA
 (1)  45A2   D3   94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45A4   7F                    4.0                   MOV     A,A             ;RETRY LINK
1879  45A5   3E   06                7.0          MVI     A,BIT1!BIT2     ;SAVE THE EXPECTED DATA
1880  45A7                                       ROUT    EDATA
 (1)  45A7   D3   95               10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45A9   7F                    4.0                   MOV     A,A             ;RETRY LINK
1881  45AA                                       ERRB    TST04X,T4CN1
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  45AA   CD   12   28        18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)         001B                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45AD   1B                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45AE   00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45AF   CD   15   28        18.0          T4CN1:. CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45B2   DA   73   45        10.0                  JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
1882                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1883  45B5                                       QMSGM   TEST05
 (1)  45B5   CD   1E   28        18.0           CALL    QUEM
 (1)         001C                               MSGN=MSGN+1
 (1)  45B8   1C                                 .BYTE   MSGN
 (1)  45B9   DA   32   46        10.0           JC      TEST05
1884                                    ;>SET THE TU78 #3 PORT SELECT SWITCH TO MASSBUS PORT 1
1885  45BC   DB   40               10.0          IN      TSTS
1886  45BE   E6   07                7.0          ANI     $07
1887  45C0   FE   05                7.0          CPI     BIT0!BIT2
1888  45C2   CA   D2   45          10.0          JZ      T4CN2
1889  45C5                                       ROUT    ADATA
 (1)  45C5   D3   94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45C7   7F                    4.0                   MOV     A,A             ;RETRY LINK
1890  45C8   3E   05                7.0          MVI     A,BIT0!BIT2
1891  45CA                                       ROUT    EDATA
 (1)  45CA   D3   95               10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45CC   7F                    4.0                   MOV     A,A             ;RETRY LINK
1892  45CD                                       ERRB    TST04X,T4CN2
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  45CD   CD   12   28        18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)         001D                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D0   1D                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45D1   00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45D2   CD   15   28        18.0          T4CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45D5   DA   73   45        10.0                  JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
1893                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
```

```
                                        QMSGM   TEST05
                                        CALL    QUEM
                                        MSGN=MSGN+1
                                        .BYTE   MSGN
                                        JC      TEST05
                                ;>SET THE TU78 #3 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
                                        IN      TSTS
                                        ANI     $07
                                        CPI     BIT0!BIT1
                                        JZ      T4CN3
                                        ROUT    ADATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MVI     A,BIT0!BIT1
                                        ROUT    EDATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    TST04X,T4CN3
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                T4CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
                                        QMSGM   TEST05
                                        CALL    QUEM
                                        MSGN=MSGN+1
                                        .BYTE   MSGN
                                        JC      TEST05
                                ;>SET THE TU78 #3 PORT SELECT SWITCH TO MAINTENANCE
                                        IN      ,,S
                                        ANI     $07
                                        CPI     BIT0!BIT1!BIT2
                                        JZ      T4CN4
                                        ROUT    ADATA
                                                OUT     ADATA           ;WRI   AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MVI     A,BIT0!BIT1!BIT2
                                        ROUT    EDATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    TST04X,T4CN4
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                T4CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
                                        ENDTST  TST04X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
```

```
(2)  461E                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
(2)  461E  CD  06  28        18.0                          CALL    REQST
(2)  4621  00                                              .BYTE                   ;DATA PATTERN NUMBER
(2)  4622  00  00                                          .WORD                   ;SYSTEM '*' COUNT
(2)  4624  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2)  4626  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
(2)  4627  07                                              .BYTE   7               ;REQUEST CODE
(1)  4628  3A  9A  4F        13.0      LDA     ITERA           ;GET ITERATION COUNT
(1)  462B  3D               4.0       DCR     A               ;DOWNCOUNT
(1)  462C  32  9A  4F        13.0      STA     ITERA           ;SAVE COUNT
(1)  462F  F2  73  45        10.0      JP      TST04X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 05 - LOOP THRESHOLD CONTROL BITS - MTA #0

                        4632    ST
                                ;******************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #0
                        4632    SD
                                ;******************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #0 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
                        4632    SP
                                ;******************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 0
                                ;*    CLEAR THE THRESHOLD DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #0 REGISTER #3
                                ;*    :   INPUT MTA #0 REGISTER #5
                                ;*    :   IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE THRESHOLD DATA
                                ;*    :   DO UNTIL THE THRESHOLD DATA=10(8)
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
                        4632    SE      .
                                ;******************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 05
                                ;*MTA2 MICRO ERROR 42
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #0
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #0 THRESHOLD NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                        4632    S
                                ;******************************************************************************
```

```
1956  4632                                  TEST05: TESTX   @05              ;SET UP THE TEST NUMBER
 (1)  4632  3E  05              7.0                        MVI     A,@05            ;DEFINE THE TEST NUMBER
 (1)  4634  CD  03  28         18.0                        CALL    TSET             ;SETUP THE TEST
1957                                   ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #0
1958                                   ;&M8954, M8955'S
1959  4637  3A  0A  4A          13.0  TST05X: LDA     UNITMP           ;GET THE UNIT MAP
1960  463A  E6  01                7.0          ANI     @001             ;TEST UNIT 0?
1961  463C  CA  90  46            10.0          JZ      TEST06           ;NO-CHECK FOR TEST06
1962                                                                     ;YES-RUN THE TEST
1963  463F  CD  70  48            18.0          CALL    CLEAR            ;CLEAN UP THE PORTS
1964  4642  DB  E0                10.0          IN      INTSTA           ;GET SELECT STATUS
1965  4644  E6  80                 7.0          ANI     BIT7
1966  4646  D3  E0                10.0          OUT     MBSEL            ;SELECT PORT 0
1967  4648  21  09  4A            10.0          LXI     H,DATA           ;LOAD ADDRESS OF DATA
1968  464B  AF                     4.0          XRA     A                ;CLEAR THE ACCUMULATOR
1969  464C  77                     7.0          MOV     M,A              ;STORE IN DATA
1970                                                                     ;
1971  464D  3E  83                 7.0  ALP01:  MVI     A,@203           ;SELECT MTA #0 REGISTER 3
1972  464F  D3  40                10.0          OUT     TCMD
1973  4651  7E                     7.0          MOV     A,M              ;GET THE THRESHOLD DATA
1974  4652  D3  4C                10.0          OUT     TCMD             ;LOAD MTA REGISTER 0
1975  4654  3E  85                 7.0          MVI     A,@205           ;SELECT MTA #0 REGISTER 5
1976  4656  D3  40                10.0          OUT     TCMD
1977  4658  7E                     7.0          MOV     A,M              ;GET THE THRESHOLD DATA
1978  4659  47                     4.0          MOV     B,A              ;SAVE FOR DATA COMPARE
1979  465A  DB  40                10.0          IN      TSTS             ;READ THE THRESHOLD REGISTER
1980  465C  E6  07                 7.0          ANI     $07              ;REMOVE THE JUNK BITS
1981  465E  B8                     4.0          CMP     B                ;COMPARE ACTUAL AND EXP.
1982  465F  CA  6E  46            10.0          JZ      ALP05            ;EQUAL - CONTINUE
1983  4662                                      ROUT    ADATA            ;STORE ACTUAL DATA
 (1)  4662  D3  94                10.0                        OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4664  7F                     4.0                        MOV     A,A              ;RETRY LINK
1984  4665  7E                     7.0          MOV     A,M              ;GET EXP. DATA
1985  4666                                      ROUT    EDATA            ;STORE EXP. DATA
 (1)  4666  D3  95                10.0                        OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  4668  7F                     4.0                        MOV     A,A              ;RETRY LINK
1986  4669                                      ERRB    ALP01,ALP05,0
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4669  CD  12  28            18.0                        CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0022                                              MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  466C  22                                                .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  466D  00                                                .BYTE   0                ;PRINT ROUTINE NUMBER
 (1)  466E  CD  15  28            18.0                ALP05:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4671  DA  4D  46            10.0                        JC      ALP01            ;LOOP ADDRESS IF LOOP SPECIFIED
1987                                   ;>DATA WRITTEN TO MTA #0 THRESHOLD NOT = DATA READ
1988  4674  7E                     7.0          MOV     A,M              ;GET DATA
1989  4675  3C                     4.0          INR     A                ;UPDATE THE DATA BYTE
1990  4676  77                     7.0          MOV     M,A              ;SAVE UPDATED DATA
1991  4677  FE  08                 7.0          CPI     @10              ;DONE?
1992  4679  C2  4D  46            10.0          JNZ     ALP01            ;NO - CONTINUE UNTIL DONE
1993
```

```
1995  467C                                          ENDTST  TST05X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  467C                                         REQ     7         ;FAKE CALL TO KEEP TEST ALIVE
 (2)  467C  CD  06  28        18.0                                CALL    REQST
 (2)  467F  00                                                    .BYTE                   ;DATA PATTERN NUMBER
 (2)  4680  00  00                                                .WORD                   ;SYSTEM '#' COUNT
 (2)  4682  00  00                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4684  00                                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4685  07                                                    .BYTE   7               ;REQUEST CODE
 (1)  4686  3A  9A  4F        13.0                LDA     ITERA       ;GET ITERATION COUNT
 (1)  4689  3D                 4.0                DCR     A           ;DOWNCOUNT
 (1)  468A  32  9A  4F        13.0                STA     ITERA       ;SAVE COUNT
 (1)  468D  F2  37  46        10.0                JP      TST05X      ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 06 - LOOP THRESHOLD CONTROL BITS - MTA #1

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #1
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #1 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 1
                                ;*    CLEAR THE THRESHOLD DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #1 REGISTER #3
                                ;*    :   INPUT MTA #1 REGISTER #5
                                ;*    :   IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE THRESHOLD DATA
                                ;*    :   DO UNTIL THE THRESHOLD DATA=10(8)
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 06
                                ;*MTA2 MICRO ERROR 43
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #1
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #1 THRESHOLD NOT - DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED - NNNN
                                S
                                ; ************************************************************************
```

```
2035  4690                                  TEST06: TESTX   @06              ;SET UP THE TEST NUMBER
 (1)  4690  3E  06              7.0                    MVI     A,@06           ;DEFINE THE TEST NUMBER
 (1)  4692  CD  03  28         18.0                    CALL    TSET            ;SETUP THE TEST
2036                                  ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #1
2037                                  ;@M8954, M8955'S
2038  4695  3A  0A  4A         13.0  TST06X: LDA     UNITMP          ;GET THE UNIT MAP
2039  4698  E6  02                7.0          ANI     @002            ;TEST UNIT 1?
2040  469A  CA  F0  46          10.0          JZ      TEST07          ;NO-CHECK FOR TEST07
2041                                                                  ;YES-RUN THE TEST
2042  469D  CD  70  48          18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
2043  46A0  DB  E0               10.0          IN      INTSTA          ;GET SELECT STATUS
2044  46A2  E6  80                7.0          ANI     BIT7
2045  46A4  F6  01                7.0          ORI     @1              ;OR IN THE PORT SELECT BITS
2046  46A6  D3  E0               10.0          OUT     MBSEL           ;SELECT PORT 1
2047  46A8  21  09  4A          10.0          LXI     H,DATA          ;LOAD ADDRESS OF DATA
2048  46AB  AF                    4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2049  46AC  77                    7.0          MOV     M,A             ;STORE IN DATA
2050                                                                  ;
2051  46AD  3E  83                7.0  ALP02:  MVI     A,@203          ;SELECT MTA #1 REGISTER 3
2052  46AF  D3  40               10.0          OUT     TCMD
2053  46B1  7E                    7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2054  46B2  D3  40               10.0          OUT     TCMD            ;LOAD MTA REGISTER 0
2055  46B4  3E  85                7.0          MVI     A,@205          ;SELECT MTA #1 REGISTER 5
2056  46B6  D3  40               10.0          OUT     TCMD
2057  46B8  7E                    7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2058  46B9  47                    4.0          MOV     B,A             ;SAVE FOR DATA COMPARE
2059  46BA  DB  40               10.0          IN      TSTS            ;READ THE THRESHOLD REGISTER
2060  46BC  E6  07                7.0          ANI     $07             ;REMOVE THE JUNK BITS
2061  46BE  B8                    4.0          CMP     B               ;COMPARE ACTUAL AND EXP.
2062  46BF  CA  CE  46           10.0          JZ      ALP06           ;EQUAL - CONTINUE
2063  46C2                                    ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  46C2  D3  94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46C4  7F                    4.0                  MOV     A,A             ;RETRY LINK
2064  46C5  7E                    7.0          MOV     A,M             ;GET EXP. DATA
2065  46C6                                    ROUT    EDATA           ;STORE EXP. DATA
 (1)  46C6  D3  95               10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46C8  7F                    4.0                  MOV     A,A             ;RETRY LINK
2066  46C9                                    ERRB    ALP02,ALP06,0
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46C9  CD  12  28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0023                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46CC  23                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46CD  00                                        .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  46CE  CD  15  28          18.0          ALP06:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46D1  DA  AD  46          10.0                  JC      ALP02           ;LOOP ADDRESS IF LOOP SPECIFIED
2067                                  ;>DATA WRITTEN TO MTA #1 THRESHOLD NOT = DATA READ
2068  46D4  7E                    7.0          MOV     A,M             ;GET DATA
2069  46D5  3C                    4.0          INR     A               ;UPDATE THE DATA BYTE
2070  46D6  77                    7.0          MOV     M,A             ;SAVE UPDATED DATA
2071  46D7  FE  08                7.0          CPI     @10             ;DONE?
2072  46D9  C2  AD  46           10.0          JNZ     ALP02           ;NO - CONTINUE UNTIL DONE
2073
```

```
2075  46DC                                        ENDTST  TST06X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46DC                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46DC  CD  06  28        18.0                                 CALL    RFQST
 (2)  46DF  00                                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  46E0  00  00                                                 .WORD                   ;SYSTEM '*' COUNT
 (2)  46E2  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46E4  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46E5  07                                                     .BYTE   7               ;REQUEST CODE
 (1)  46E6  3A  9A  4F        13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1)  46E9  3D                 4.0                 DCR     A               ;DOWNCOUNT
 (1)  46EA  32  9A  4F        13.0                 STA     ITERA           ;SAVE COUNT
 (1)  46ED  F2  95  46        10.0                 JP      TST06X          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 07 - LOOP THRESHOLD CONTROL BITS - MTA #2

                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #2
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #2 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #2 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 2
                                ;*     CLEAR THE THRESHOLD DATA
                                ;*     BGNDO
                                ;*     :   ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #2 REGISTER #3
                                ;*     :   INPUT MTA #2 REGISTER #5
                                ;*     :   IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR
                                ;*     :   ENDIF
                                ;*     :   INCREMENT THE THRESHOLD DATA
                                ;*     :   DO UNTIL THE THRESHOLD DATA=10(8)
                                ;*     ENDDO
                                ;*     CALL SUBROUTINE CLEAR
                                ;*ENDTST
                                SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 07
                                ;*MTA2 MICRO ERROR 44
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #2
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #2 THRESHOLD NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                S
                                ;*********************************************************************
```

```
                                TEST07: TESTX   @07             ;SET UP THE TEST NUMBER
                                                MVI     A,@07           ;DEFINE THE TEST NUMBER
                                                CALL    TSET            ;SETUP THE TEST
                                ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #2
                                ;&M8954, M8955'S
                                TST07X: LDA     UNITMP          ;GET THE UNIT MAP
                                        ANI     @004            ;TEST UNIT 2?
                                        JZ      TEST10          ;NO-CHECK FOR TEST10
                                                                ;YES-RUN THE TEST
                                        CALL    CLEAR           ;CLEAN UP THE PORTS
                                        IN      INTSTA          ;GET SELECT STATUS
                                        ANI     BIT7
                                        ORI     @2
                                        OUT     MBSEL           ;SELECT PORT 2
                                        LXI     H,DATA          ;LOAD ADDRESS OF DATA
                                        XRA     A               ;CLEAR THE ACCUMULATOR
                                        MOV     M,A             ;STORE IN DATA
                                                                ;
                                ALP03:  MVI     A,@203          ;SELECT MTA #2 REGISTER 3
                                        OUT     TCMD
                                        MOV     A,M             ;GET THE THRESHOLD DATA
                                        OUT     TCMD            ;LOAD MTA REGISTER 0
                                        MVI     A,@205          ;SELECT MTA #2 REGISTER 5
                                        OUT     TCMD
                                        MOV     A,M             ;GET THE THRESHOLD DATA
                                        MOV     B,A             ;SAVE FOR DATA COMPARE
                                        IN      TSTS            ;READ THE THRESHOLD REGISTER
                                        ANI     $07             ;REMOVE THE JUNK BITS
                                        CMP     B               ;COMPARE ACTUAL AND EXP.
                                        JZ      ALP07           ;EQUAL - CONTINUE
                                        ROUT    ADATA           ;STORE ACTUAL DATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MOV     A,M             ;GET EXP. DATA
                                        ROUT    EDATA           ;STORE EXP. DATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    ALP03,ALP07,0
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE   0               ;PRINT ROUTINE NUMBER
                                        ALP07:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      ALP03           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>DATA WRITTEN TO MTA #2 THRESHOLD NOT = DATA READ
                                        MOV     A,M             ;GET DATA
                                        INR     A               ;UPDATE THE DATA BYTE
                                        MOV     M,A             ;SAVE UPDATED DATA
                                        CPI     @10             ;DONE?
                                        JNZ     ALP03           ;NO - CONTINUE UNTIL DONE
```

```
2155  473C                                        ENDTST  TST07X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  473C                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  473C  CD  06  28        18.0                                 CALL    REQST
 (2)  473F  00                                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  4740  00  00                                                 .WORD                   ;SYSTEM "*" COUNT
 (2)  4742  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4744  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4745  07                                                     .BYTE   7               ;REQUEST CODE
 (1)  4746  3A  9A  4F        13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1)  4749  3D                 4.0                 DCR     A               ;DOWNCOUNT
 (1)  474A  32  9A  4F        13.0                 STA     ITERA           ;SAVE COUNT
 (1)  474D  F2  F5  46        10.0                 JP      TST07X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - LOOP THRESHOLD CONTROL BITS - MTA #3

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #3
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #3 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 3
                                ;*    CLEAR THE THRESHOLD DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #3 REGISTER #3
                                ;*    :   INPUT MTA #3 REGISTER #5
                                ;*    :   IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE THRESHOLD DATA
                                ;*    :   DO UNTIL THE THRESHOLD DATA=10(8)
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 10
                                ;*MTA2 MICRO ERROR 45
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #3
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #3 THRESHOLD NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                S
                                ; ************************************************************************
```

```
2195  4750                                  TEST10: TESTX   @10              ;SET UP THE TEST NUMBER
 (1)  4750  3E  08              7.0                    MVI     A,@10           ;DEFINE THE TEST NUMBER
 (1)  4752  CD  03  28         18.0                    CALL    TSET            ;SETUP THE TEST
2196                                  ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #3
2197                                  ;&M8954, M8955'S
2198  4755  3A  0A  4A          13.0  TST10X: LDA     UNITMP          ;GET THE UNIT MAP
2199  4758  E6  08                7.0          ANI     @010            ;TEST UNIT 3?
2200  475A  CA  B0  47          10.0          JZ      TEST11          ;NO-CHECK FOR TEST11
2201                                                                  ;YES-RUN THE TEST
2202  475D  CD  70  48          18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
2203  4760  DB  E0                10.0          IN      INTSTA          ;GET SELECT STATUS
2204  4762  E6  80                 7.0          ANI     BIT7
2205  4764  F6  03                 7.0          ORI     @3
2206  4766  D3  E0                10.0          OUT     MBSEL           ;SELECT PORT 3
2207  4768  21  09  4A            10.0          LXI     H,DATA          ;LOAD ADDRESS OF DATA
2208  476B  AF                      4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2209  476C  77                      7.0          MOV     M,A             ;STORE IN DATA
2210                                                                  ;
2211  476D  3E  83                 7.0  ALP04:  MVI     A,@203          ;SELECT MTA #3 REGISTER 3
2212  476F  D3  40                10.0          OUT     TCMD
2213  4771  7E                     7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2214  4772  D3  40                10.0          OUT     TCMD            ;LOAD MTA REGISTER 0
2215  4774  3E  85                 7.0          MVI     A,@205          ;SELECT MTA #3 REGISTER 5
2216  4776  D3  40                10.0          OUT     TCMD
2217  4778  7E                     7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2218  4779  47                     4.0          MOV     B,A             ;SAVE FOR DATA COMPARE
2219  477A  DB  40                10.0          IN      TSTS            ;READ THE THRESHOLD REGISTER
2220  477C  E6  07                 7.0          ANI     $07             ;REMOVE THE JUNK BITS
2221  477E  B8                      4.0          CMP     B               ;COMPARE ACTUAL AND EXP.
2222  477F  CA  8E  47            10.0          JZ      ALP08           ;EQUAL - CONTINUE
2223  4782                                      ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4782  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4784  7F                     4.0                  MOV     A,A             ;RETRY LINK
2224  4785  7E                     7.0          MOV     A,M             ;GET EXP. DATA
2225  4786                                      ROUT    EDATA           ;STORE EXP. DATA
 (1)  4786  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4788  7F                     4.0                  MOV     A,A             ;RETRY LINK
2226  4789                                      ERRB    ALP04,ALP08,0
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4789  CD  12  28         18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0025                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  478C  25                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  478D  00                                      .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  478E  CD  15  28         18.0          ALP08:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4791  DA  6D  47         10.0                  JC      ALP04           ;LOOP ADDRESS IF LOOP SPECIFIED
2227                                  ;>DATA WRITTEN TO MTA #3 THRESHOLD NOT = DATA READ
2228  4794  7E                     7.0          MOV     A,M             ;GET DATA
2229  4795  3C                     4.0          INR     A               ;UPDATE THE DATA BYTE
2230  4796  77                     7.0          MOV     M,A             ;SAVE UPDATED DATA
2231  4797  FE  08                 7.0          CPI     @10             ;DONE?
2232  4799  C2  6D  47            10.0          JNZ     ALP04           ;NO - CONTINUE UNTIL DONE
2233
```

```
2235  479C                                        ENDTST  TST10X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  479C                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  479C  CD  06  28        18.0                                CALL    REQST
 (2)  479F  00                                                    .BYTE                   ;DATA PATTERN NUMBER
 (2)  47A0  00  00                                                .WORD                   ;SYSTEM '*' COUNT
 (2)  47A2  00  00                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47A4  00                                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  47A5  07                                                    .BYTE   7               ;REQUEST CODE
 (1)  47A6  3A  9A  4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  47A9  3D                 4.0            DCR     A               ;DOWNCOUNT
 (1)  47AA  32  9A  4F        13.0            STA     ITERA           ;SAVE COUNT
 (1)  47AD  F2  55  47        10.0            JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 11 - COMMAND FUNCTION TEST - MTA #0
47B0                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*COMMAND FUNCTION TEST MTA #0
47B0                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                                ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                                ;*SET AND/OR CLEAR.
47B0                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #0 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #0
                                ;*  CALL SUBROUTINE CMDCOM
                                ;*ENDTST
47B0                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA2 MICRO TEST 11
                                ;*MTA2 MICRO ERROR 47
                                ;*MTA COMMAND FUNCTION TEST - MTA #0
                                ;*M8954, M8955'S
                                ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                                ;*ACTUAL = NNNN
                                ;*
                                ;*MTA2 MICRO TEST 11
                                ;*MTA2 MICRO ERROR 50
                                ;*MTA COMMAND FUNCTION TEST - MTA #0
                                ;*M8954, M8955'S
                                ;*'READY ON'' IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 11
                                ;*MTA2 MICRO ERROR 51
                                ;*MTA COMMAND FUNCTION TEST - MTA #0
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 11
                                ;*MTA2 MICRO ERROR 52
                                ;*MTA COMMAND FUNCTION TEST - MTA #0
                                ;*M8954, M8955'S
                                ;*'PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 53
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*'PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 54
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*'PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 55
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*'PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 56
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*'DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 57
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*'FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 60
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*'RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 61
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 62
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*''TU78'' IN MTA REGISTER 2 NOT SET
 2327  4780             S
  (1)                   ; ************************************************************************************
```

```
                                TEST11: TESTX   @011            ;INITIALIZE THE TEST
                                                MVI     A,@011          ;DEFINE THE TEST NUMBER
                                                CALL    TSET            ;SETUP THE TEST
                                ;%MTA COMMAND FUNCTION TEST - MTA #0
                                ;&M8954, M8955'S
                                        LDA     UNITMP          ;GET THE UNIT MAP
                                        ANI     @001            ;TEST UNIT 0
                                        JZ      TEST12          ;NO-GO CHECK TEST 12
                                TST11X: CALL    CLEAR           ;CLEAN UP THE PORTS
                                        IN      INTSTA          ;GET MASSBUS SELECT
                                        ANI     BIT7
                                        OUT     MBSEL           ;SELECT TU PORT #0
                                        CALL    CMDCOM
                                        ENDTST  TST11X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
                                                CALL    REQST
                                                .BYTE                   ;DATA PATTERN NUMBER
                                                .WORD                   ;SYSTEM '*' COUNT
                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                .BYTE                   ;DATA COMPARE FLAG IF =1
                                                .BYTE   7               ;REQUEST CODE
                                        LDA     ITERA           ;GET ITERATION COUNT
                                        DCR     A               ;DOWNCOUNT
                                        STA     ITERA           ;SAVE COUNT
                                        J?      TST11X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - COMMAND FUNCTION TEST - MTA #1
47DD                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*COMMAND FUNCTION TEST MTA #1
47DD                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                                ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                                ;*SET AND/OR CLEAR.
47DD                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #1 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #1
                                ;*  CALL SUBROUTINE CMDCOM
                                ;*ENDTST
47DD                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 47
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                                ;*ACTUAL = NNNN
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 50
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''READY ON'' IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 51
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REG. 2 NOT 0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRC TEST 12
                                ;*MTA2 MICRO ERROR 52
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
2384                            ;*
2385                            ;*MTA2 MICRO TEST 12
2386                            ;*MTA2 MICRO ERROR 53
2387                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2388                            ;*M8954, M8955'S
2389                            ;*'PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2390                            ;*
2391                            ;*MTA2 MICRO TEST 12
2392                            ;*MTA2 MICRO ERROR 54
2393                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2394                            ;*M8954, M8955'S
2395                            ;*'PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
2396                            ;*
2397                            ;*MTA2 MICRO TEST 12
2398                            ;*MTA2 MICRO ERROR 55
2399                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2400                            ;*M8954, M8955'S
2401                            ;*'PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
2402                            ;*
2403                            ;*MTA2 MICRO TEST 12
2404                            ;*MTA2 MICRO ERROR 56
2405                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2406                            ;*M8954, M8955'S
2407                            ;*'DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2408                            ;*
2409                            ;*MTA2 MICRO TEST 12
2410                            ;*MTA2 MICRO ERROR 59
2411                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2412                            ;*M8954, M8955'S
2413                            ;*'FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2414                            ;*
2415                            ;*MTA2 MICRO TEST 12
2416                            ;*MTA2 MICRO ERROR 60
2417                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2418                            ;*M8954, M8955'S
2419                            ;*'RDY ON' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
2420                            ;*
2421                            ;*MTA2 MICRO TEST 12
2422                            ;*MTA2 MICRO ERROR 61
2423                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2424                            ;*M8954, M8955'S
2425                            ;*''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
2426                            ;*
2427                            ;*MTA2 MICRO TEST 12
2428                            ;*MTA2 MICRO ERROR 62
2429                            ;*MTA COMMAND FUNCTION TEST - MTA #1
2430                            ;*M8954, M8955'S
2431                            ;*''TU78'' IN MTA REGISTER 2 NOT SET
2432  47DD                      S
 (1)                            ; ***************************************************************************
```

```
2434  47DD                                TEST12: TESTX   @012            ;INITIALIZE THE TEST
 (1)  47DD  3E  0A              7.0                      MVI     A,@012          ;DEFINE THE TEST NUMBER
 (1)  47DF  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
2435                                    ;%MTA COMMAND FUNCTION TEST - MTA #1
2436                                    ;&M8954, M8955'S
2437  47E2  3A  0A  4A         13.0           LDA     UNITMP          ;GET THE UNIT MAP
2438  47E5  E6  02              7.0           ANI     @002            ;TEST UNIT 1
2439  47E7  CA  0C  48         10.0           JZ      TEST13          ;NO-GO CHECK TEST 13
2440  47EA  CD  70  48         18.0   TST12X: CALL    CLEAR           ;CLEAN UP THE PORTS
2441  47ED  DB  E0             10.0           IN      INTSTA          ;GET MASSBUS SELECT
2442  47EF  E6  80              7.0           ANI     BIT7
2443  47F1  F6  01              7.0           ORI     $01
2444  47F3  D3  E0             10.0           OUT     MBSEL           ;SELECT TU PORT #1
2445  47F5  CD  9F  48         18.0           CALL    CMDCOM
2446  47F8                                   ENDTST  TST12X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47F8                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47F8  CD  06  28         18.0                      CALL    REQST
 (2)  47FB  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  47FC  00  00                                       .WORD                   ;SYSTEM '#' COUNT
 (2)  47FE  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4800  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4801  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4802  3A  9A  4F         13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  4805  3D              4.0               DCR     A               ;DOWNCOUNT
 (1)  4806  32  9A  4F         13.0           STA     ITERA           ;SAVE COUNT
 (1)  4809  F2  EA  47         10.0           JP      TST12X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - COMMAND FUNCTION TEST - MTA #2
480C                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*COMMAND FUNCTION TEST MTA #2
480C                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                                ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                                ;*SET AND/OR CLEAR.
480C                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #2 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #2
                                ;*  CALL SUBROUTINE CMDCOM
                                ;*ENDTST
480C                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 47
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                                ;*ACTUAL = NNNN
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 50
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'READY ON'' IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 51
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 52
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 53
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 54
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 55
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 56
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 57
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 60
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*'RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 61
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 62
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*''TU78'' IN MTA REGISTER 2 NOT SET
                                ;
480C                            S
                                ; ********************************************************************************
```

```
2540  480C                                  TEST13: TESTX   @013            ;INITIALIZE THE TEST
 (1)  480C  3E  0B              7.0                    MVI     A,@013          ;DEFINE THE TEST NUMBER
 (1)  480E  CD  03  28         18.0                    CALL    TSET            ;SETUP THE TEST
2541                                   ;%MTA COMMAND FUNCTION TEST - MTA #2
2542                                   ;@M8954, M8955'S
2543  4811  3A  0A  4A         13.0            LDA     UNITMP          ;GET THE UNIT MAP
2544  4814  E6  04               7.0            ANI     @004            ;TEST UNIT 2
2545  4816  CA  3B  48         10.0            JZ      TEST14          ;NO-GO CHECK TEST 14
2546  4819  CD  70  48         18.0  TST13X:   CALL    CLEAR           ;CLEAN UP THE PORTS
2547  481C  DB  E0              10.0            IN      INTSTA          ;GET MASSBUS SELECT
2548  481E  E6  80               7.0            ANI     BIT7
2549  4820  F6  02               7.0            ORI     $02
2550  4822  D3  E0              10.0            OUT     MBSEL           ;SELECT TU PORT #2
2551  4824  CD  9F  48         18.0            CALL    CMDCOM
2552  4827                                    ENDTST  TST13X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4827                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4827  CD  06  28         18.0                    CALL    REQST
 (2)  482A  00                                         .BYTE                   ;DATA PATTERN NUMBER
 (2)  482B  00  00                                     .WORD                   ;SYSTEM '@' COUNT
 (2)  482D  00  00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  482F  00                                         .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4830  07                                         .BYTE   7               ;REQUEST CODE
 (1)  4831  3A  9A  4F         13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  4834  3D               4.0            DCR     A               ;DOWNCOUNT
 (1)  4835  32  9A  4F         13.0            STA     ITERA           ;SAVE COUNT
 (1)  4838  F2  19  48         10.0            JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - COMMAND FUNCTION TEST - MTA #3
                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*COMMAND FUNCTION TEST MTA #3
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                                ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                                ;*SET AND/OR CLEAR.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   IF TU PORT #3 NOT SELECTED BY USER
                                ;*   :  THEN-NEXT TEST
                                ;*   :  ELSE-CONTINUE
                                ;*   ENDIF
                                ;*   CALL SUBROUTINE CLEAR
                                ;*   SELECT TU PORT #3
                                ;*   CALL SUBROUTINE CMDCOM
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 47
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                                ;*ACTUAL = NNNN
                                ;*
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 50
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*'READY ON'' IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 51
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 52
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*'PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
2596                                    ;*
2597                                    ;*MTA2 MICRO TEST 14
2598                                    ;*MTA2 MICRO ERROR 53
2599                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2600                                    ;*M8954, M8955'S
2601                                    ;*''PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2602                                    ;*
2603                                    ;*MTA2 MICRO TEST 14
2604                                    ;*MTA2 MICRO ERROR 54
2605                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2606                                    ;*M8954, M8955'S
2607                                    ;*''PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
2608                                    ;*
2609                                    ;*MTA2 MICRO TEST 14
2610                                    ;*MTA2 MICRO ERROR 55
2611                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2612                                    ;*M8954, M8955'S
2613                                    ;*''PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
2614                                    ;*
2615                                    ;*MTA2 MICRO TEST 14
2616                                    ;*MTA2 MICRO ERROR 56
2617                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2618                                    ;*M8954, M8955'S
2619                                    ;*''DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2620                                    ;*
2621                                    ;*MTA2 MICRO TEST 14
2622                                    ;*MTA2 MICRO ERROR 57
2623                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2624                                    ;*M8954, M8955'S
2625                                    ;*''FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2626                                    ;*
2627                                    ;*MTA2 MICRO TEST 14
2628                                    ;*MTA2 MICRO ERROR 60
2629                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2630                                    ;*M8954, M8955'S
2631                                    ;*''RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
2632                                    ;*
2633                                    ;*MTA2 MICRO TEST 14
2634                                    ;*MTA2 MICRO ERROR 61
2635                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2636                                    ;*M8954, M8955'S
2637                                    ;*''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
2638                                    ;*
2639                                    ;*MTA2 MICRO TEST 14
2640                                    ;*MTA2 MICRO ERROR 62
2641                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2642                                    ;*M8954, M8955'S
2643                                    ;*''TU78'' IN MTA REGISTER 2 NOT SET
2644  4838                              S
 (1)                                    ; ************************************************************************************
```

```
2646  4838                                  TEST14: TESTX   @014          ;INITIALIZE THE TEST
 (1)  4838  3E  0C                    7.0                 MVI     A,@014          ;DEFINE THE TEST NUMBER
 (1)  483D  CD  03  28               18.0                 CALL    TSET            ;SETUP THE TEST
2647                                        ;%MTA COMMAND FUNCTION TEST - MTA #3
2648                                        ;&M8954, M8955'S
2649  4840  3A  0A  4A               13.0          LDA     UNITMP          ;GET THE UNIT MAP
2650  4843  E6  08                    7.0          ANI     @010            ;TEST UNIT 3
2651  4845  CA  6A  48               10.0          JZ      TEST15          ;NO-GO CHECK TEST 15
2652  4848  CD  70  48               18.0  TST14X: CALL    CLEAR           ;CLEAN UP THE PORTS
2653  484B  DB  E0                   10.0          IN      INTSTA          ;GET MASSBUS SELECT
2654  484D  E6  80                    7.0          ANI     BIT7
2655  484F  F6  03                    7.0          ORI     $03
2656  4851  D3  E0                   10.0          OUT     MBSEL           ;SELECT TU PORT #3
2657  4853  CD  9F  48               18.0          CALL    CMDCOM
2658  4856                                        ENDTST  TST14X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4856                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4856  CD  06  28               18.0                 CALL    REQST
 (2)  4859  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  485A  00  00                                        .WORD                   ;SYSTEM '' OUNT
 (2)  485C  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  485E  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  485F  07                                            .BYTE   7               ;REQUEST CODE
 (1)  4860  3A  9A  4F               13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4863  3D                        4.0          DCR     A               ;DOWNCOUNT
 (1)  4864  32  9A  4F               13.0          STA     ITERA           ;SAVE COUNT
 (1)  4867  F2  48  48               10.0          JP      TST14X          ;DO TEST UNTIL TILL = 0
2659
2660  486A  C3  6D  48               10.0  TEST15: JMP     EXIT
2661  486D  C3  18  28               10.0  EXIT:   JMP     TSTEND
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                SSUB
                                ;*************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CLEAR ALL TU PORTS
                                SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
                                SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*      SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*      CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*      CLEAR PORT SELECT FOR TRANSPORT
                                ;*      CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*      CLEAR PORT DIAGNOSTIC CONTROL
                                ;*      CLEAR PORT AMTIE WORD
                                ;*ENDSUB
                                S
                                ;*************************************************************************
 4870                   12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
 4871                   12.0          PUSH    B
 4872  06  00            7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
 4874  DB  E0           10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
 4876  E6  80            7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
 4878  B0                4.0          ORA     B               ;ADD IN THE SELECTED PORT #
 4879  D3  E0           10.0          OUT     MBSEL           ;RESET TO THIS PORT
 487B  3E  80            7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
 487D  D3  40           10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
 487F  AF                4.0          XRA     A               ;CLEAR TU COMMAND A
 4880  D3  40           10.0          OUT     TCMD            ;
 4882  3E  81            7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
 4884  D3  40           10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
 4886  3E  00            7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
 4888  D3  40           10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
 488A  3E  03            7.0          MVI     A,EOTCLR        ;LOAD TU ''EOT STATUS CLEAR'' COMMAND
 488C  D3  40           10.0          OUT     TCMD            ;ISSUE TU ''EOT STATUS CLEAR'' COMMAND
 488E  AF                4.0          XRA     A
 488F  D3  44           10.0          OUT     TAMT            ;CLEAR AMTIE WORD
 4891  D3  48           10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
 4893  D3  4C           10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
 4895  04                4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
 4896  78                4.0          MOV     A,B
```

```
2704  4897  FE  04        7.0   CPI   4      ;DONE?
2705  4899  C2  74  48   10.0   JNZ   CLRLP  ;NO - CLEAR THIS PORT ALSO
2706  489C  C1           10.0   POP   B      ;RESET B & C
2707  489D  F1           10.0   POP   PSW    ;ALL DONE
2708  489E  C9           10.0   RET          ;EXIT
```

```
2710                                    .SBTTL  SUBROUTINE COMMAND COMMON
2711  489F                              SSUB
 (1)                                    ;*****************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
2712                                    ;*COMMAND COMMON CODE
2713  489F                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2714                                    ;*THIS SUBROUTINE IS USED TO TEST THE FUNCTIONALITY OF AN MTA BOARD
2715                                    ;*COMMAND REPERTOIRE BY ISSUING THE CLEAR TU, SET PE, SET GCR, DSE, AND
2716                                    ;*TEST COMMANDS AND MONITORING THE RESPONSE.
2717  489F                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2718                                    ;*BGNSUB
2719                                    ;*   IF MANUAL INTERVENTION NOT ALLOWED
2720                                    ;*   :  THEN - EXIT  HE SUBROUTINE
2721                                    ;*   :  ELSE - CONTINUE
2722                                    ;*   ENDIF
2723                                    ;*   ASK THE USER TO PUT THE TU78 OFF-LINE
2724                                    ;*   SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS
2725                                    ;*   ISSUE MTA CLEAR TU COMMAND
2726                                    ;*   IF MTA STATUS BITS FWD, REV, WRT, WRT INH, LWR, DSE, CMD PE, AND RDY ON=0
2727                                    ;*   :  THEN-CONTINUE
2728                                    ;*   :  ELSE-ERROR
2729                                    ;*   ENDIF
2730                                    ;*   ISSUE MTA SET PE COMMAND
2731                                    ;*   IF MTA STATUS BITS PES AND PEC=1
2732                                    ;*   :  THEN-CONTINUE
2733                                    ;*   :  ELSE-ERROR
2734                                    ;*   ENDIF
2735                                    ;*   ISSUE MTA SET GCR COMMAND
2736                                    ;*   IF MTA STATUS BITS PES AND PEC=0
2737                                    ;*   :  THEN-CONTINUE
2738                                    ;*   :  ELSE-ERROR
2739                                    ;*   ENDIF
2740                                    ;*   ISSUE MTA DSE COMMAND
2741                                    ;*   IF MTA STATUS BITS DSE AND FWD=1
2742                                    ;*   :  THEN-CONTINUE
2743                                    ;*   :  ELSE-ERROR
2744                                    ;*   ENDIF
2745                                    ;*   ISSUE MTA TEST COMMAND
2746                                    ;*   IF RDY ON AND CMD PE-1
2747                                    ;*   :  THEN-CONTINUE
2748                                    ;*   :  ELSE-ERROR
2749                                    ;*   ENDIF
2750                                    ;*   IF MTA STATUS BIT TU78=1
2751                                    ;*   :  THEN-CONTINUE
2752                                    ;*   :  ELSE-ERROR
2753                                    ;*   ENDIF
2754                                    ;*   CALL SUBROUTINE CLEAR
```

```
 2755                                   ;*ENDSUB
 2756  489F                             S
  (1)                                   ; **********************************************************************************
```

```
2758  489F                                    CMDCOM: QMSGM   OFFOUT          ;ASK THE USER TO PUT THE DRIVE OFF-LINE
 (1)  489F  CD  1E  28          18.0                 CALL    QUEM
 (1)        0026                                     MSGN=MSGN+1
 (1)  48A2  26                                       .BYTE   MSGN
 (1)  48A3  DA  05  4A          10.0                 JC      OFFOUT
2759                                        ;>PUT THE TU78 OFF LINE
2760  48A6  3E  80                7.0      CMDCM:  MVI     A,$80           ;LOAD MTA REGISTER 0 SELECT CODE
2761  48A8  D3  40               10.0              OUT     TCMD
2762  48AA  DB  40               10.0              IN      TSTS            ;GET TU STATUS
2763  48AC  E6  20                7.0              ANI     T.ONL           ;IS ONLINE BIT SET
2764  48AE  CA  B9  48           10.0              JZ      CMDCN0          ;NO , GOOD CONTINUE
2765  48B1                                        ROUT    ADATA           ;YES , ERROR
 (1)  48B1  D3  94               10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  48B3  7F                    4.0                      MOV     A,A             ;RETRY LINK
2766  48B4                                        ERRA    CMDCOM,CMDCN0
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  48B4  CD  0F  28          18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0027                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48B7  27                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48B8  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48B9  CD  15  28          18.0                 CMDCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48BC  DA  9F  48          10.0                            JC      CMDCOM          ;LOOP ADDRESS IF LOOP SPECIFIED
2767                                        ;>''ONL'' BIT SET IN MTA 0
2768  48BF                                    QMSGM   OFFOUT
 (1)  48BF  CD  1E  28          18.0                 CALL    QUEM
 (1)        0028                                     MSGN=MSGN+1
 (1)  48C2  28                                       .BYTE   MSGN
 (1)  48C3  DA  05  4A          10.0                 JC      OFFOUT
2769                                        ;>PUT THE TU78 ONLINE
2770  48C6  DB  40               10.0              IN      TSTS            ;GET TU STATUS
2771  48C8  E6  20                7.0              ANI     T.ONL           ;TU78 ONLINE ?
2772  48CA  C2  D5  48           10.0              JNZ     CMDCNZ          ;YES , CONTINUE
2773  48CD                                        ROUT    ADATA           ;NO , ERROR TAPE SHOULD BE ONLINE
 (1)  48CD  D3  94               10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  48CF  7F                    4.0                      MOV     A,A             ;RETRY LINK
2774  48D0                                        ERRA    CMDCOM,CMDCNZ
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  48D0  CD  0F  28          18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0029                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48D3  29                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48D4  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48D5  CD  15  28          18.0                 CMDCNZ::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48D8  DA  9F  48          10.0                            JC      CMDCOM          ;LOOP ADDRESS IF LOOP SPECIFIED
2775                                        ;>''ONL'' BIT NOT SET IN MTA 0
2776  48DB  3E  7C                7.0              MVI     A,$7C           ;SET THE MTA REGISTER 0 COMMAND BITS
2777  48DD  D3  40               10.0              OUT     TCMD
2778
2779  48DF  3E  81                7.0              MVI     A,$81           ;LOAD MTA REGISTER 1 SELECT CODE
2780  48E1  D3  40               10.0              OUT     TCMD
2781  48E3  AF                    4.0              XRA     A               ;ISSUE CLEAR TU COMMAND
2782  48E4  D3  40               10.0              OUT     TCMD
2783
2784  48E6  DB  40               10.0              IN      TSTS            ;READ MTA REGISTER 1
2785  48E8  E6  FD                7.0              ANI     $FD             ;REMOVE JUNK
```

```
2786  48EA  CA  F5  48      10.0          JZ      CMDCN1              ;=0? - CONTINUE
2787  48ED                                ROUT    ADATA               ;SAVE THE ACTUAL DATA
 (1)  48ED  D3  94          10.0                   OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  48EF  7F               4.0                   MOV     A,A                 ;RETRY LINK
2788  48F0                                ERRA    CMDCM,CMDCN1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  48F0  CD  0F  28      18.0                           CALL    ERLPA               ;PROCESS ERROR - DO 2.3
 (1)        002A                                         MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48F3  2A                                           .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  48F4  00                                           .BYTE                       ;PRINT ROUTINE NUMBER
 (1)  48F5  CD  15  28      18.0                   CMDCN1::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48F8  DA  A6  48      10.0                           JC      CMDCM               ;LOOP ADDRESS IF LOOP SPECIFIED
2789                            ;>MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
2790
2791  48FB  3E  80           7.0          MVI     A,$80               ;LOAD MTA REGISTER 0 SELECT CODE
2792  48FD  D3  40          10.0          OUT     TCMD
2793  48FF  DB  40          10.0          IN      TSTS                ;READ MTA REGISTER 0
2794  4901  E6  40           7.0          ANI     $40
2795  4903  CA  08  49      10.0          JZ      CMDCN2
2796  4906                                ERR     CMDCM,CMDCN2
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4906  CD  09  28      18.0                   CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0028                                 MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4909  28                                   .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  490A  0C                                   .BYTE
 (1)  490B  CD  15  28      18.0           CMDCN2::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  490E  DA  A6  48      10.0                   JC      CMDCM               ;LOOP ADDRESS IF LOOP SPECIFIED
2797                            ;>"READY ON" IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
2798  4911  3E  82           7.0          MVI     A,$82               ;LOAD MTA REGISTER 2 SELECT CODE
2799  4913  D3  40          10.0          OUT     TCMD
2800  4915  DB  40          10.0          IN      TSTS
2801  4917  E6  80           7.0          ANI     $80
2802  4919  CA  21  49      10.0          JZ      CMDCN3
2803  491C                                ERR     CMDCM,CMDCN3
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  491C  CD  09  28      18.0                   CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)        002C                                 MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  491F  2C                                   .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  4920  00                                   .BYTE
 (1)  4921  CD  15  28      18.0           CMDCN3::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4924  DA  A6  48      10.0                   JC      CMDCM               ;LOOP ADDRESS IF LOOP SPECIFIED
2804                            ;>"CMD PE" IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
```

```
2806  4927  3E  81        7.0  CMDCOA: MVI   A,$81          ;LOAD MTA REGISTER 1 SELECT CODE
2807  4929  D3  40       10.0          OUT   TCMD
2808  492B  3E  01        7.0          MVI   A,$01          ;ISSUE SET PE COMMAND
2809  492D  D3  40       10.0          OUT   TCMD
2810
2811  492F  3E  82        7.0          MVI   A,$82          ;LOAD MTA REGISTER 2 SELECT CODE
2812  4931  D3  40       10.0          OUT   TCMD
2813  4933  DB  40       10.0          IN    TSTS
2814  4935  E6  40        7.0          ANI   $40
2815  4937  C2  3F  49   10.0          JNZ   CMDCN4
2816  493A                             ERR   CMDCOA,CMDCN4
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  493A  CD  09  28   18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)      002D                          MSGN  =     MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  493D  2D                          .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  493E  00                          .BYTE
 (1)  493F  CD  15  28   18.0    CMDCN4::      CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4942  DA  27  49   10.0                JC    CMDCOA          ;LOOP ADDRESS IF LOOP SPECIFIED
2817                        ;>'PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2818  4945  3E  82        7.0          MVI   A,$82
2819  4947  D3  40       10.0          OUT   TCMD
2820  4949  DB  40       10.0          IN    TSTS
2821  494B  E6  40        7.0          ANI   $40
2822  494D  C2  55  49   10.0          JNZ   CMDCN5
2823  4950                             ERR   CMDCOA,CMDCN5
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4950  CD  09  28   18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)      002E                          MSGN  =     MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4953  2E                          .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4954  00                          .BYTE
 (1)  4955  CD  15  28   18.0    CMDCN5::      CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4958  DA  27  49   10.0                JC    CMDCOA          ;LOOP ADDRESS IF LOOP SPECIFIED
2824                        ;>'PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2825  495B  3E  81        7.0  CMDCOB: MVI   A,$81          ;LOAD MTA REGISTER 1 SELECT CODE
2826  495D  D3  40       10.0          OUT   TCMD
2827  495F  3E  02        7.0          MVI   A,$02          ;ISSUE SET GCR COMMAND
2828  4961  D3  40       10.0          OUT   TCMD
2829
2830  4963  3E  80        7.0          MVI   A,$80          ;LOAD MTA REGISTER 0 SELECT CODE
2831  4965  D3  40       10.0          OUT   TCMD
2832  4967  DB  40       10.0          IN    TSTS
2833  4969  E6  08        7.0          ANI   $08
2834  496B  CA  73  49   10.0          JZ    CMDCN6
```

```
2836  496E                                          ERR     CMDCOB,CMDCN6
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  496E  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        002F                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4971  2F                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4972  00                                              .BYTE
 (1)  4973  CD  15  28        18.0                  CMDCN6::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4976  DA  58  49        10.0                          JC      CMDCOB          ;LOOP ADDRESS IF LOOP SPECIFIED
2837                                        ;>'PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
2838  4979  3E  82               7.0          MVI     A,$82
2839  497B  D3  40              10.0          OUT     TCMD
2840  497D  DB  40              10.0          IN      TSTS
2841  497F  E6  40               7.0          ANI     $40
2842  4981  CA  89  49          10.0          JZ      CMDCN7
2843  4984                                    ERR     CMDCOB,CMDCN7
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4984  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0030                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4987  30                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4988  00                                              .BYTE
 (1)  4989  CD  15  28        18.0                  CMDCN7::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  498C  DA  58  49        10.0                          JC      CMDCOB          ;LOOP ADDRESS IF LOOP SPECIFIED
2844                                        ;>'PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
2845  498F  3E  81               7.0  CMDCOC: MVI     A,$81           ;LOAD MTA REG. 1 SELECT CODE
2846  4991  D3  40              10.0          OUT     TCMD
2847  4993  3E  06               7.0          MVI     A,$06           ;ISSUE THE DSE COMMAND
2848  4995  D3  40              10.0          OUT     TCMD
2849  4997  DB  40              10.0          IN      TSTS
2850  4999  E6  01               7.0          ANI     $01
2851  499B  C2  93  49          10.0          JNZ     CMDCN8
2852  499E                                    ERR     CMDCOC,CMDCN8
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  499E  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0031                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49A1  31                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49A2  00                                              .BYTE
 (1)  49A3  CD  15  28        18.0                  CMDCN8::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  49A6  DA  8F  49        10.0                          JC      CMDCOC          ;LOOP ADDRESS IF LOOP SPECIFIED
2853                                        ;>'DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2854
2855  49A9  DB  40              10.0          IN      TSTS
2856  49AB  E6  40               7.0          ANI     $40
2857  49AD  C2  B5  49          10.0          JNZ     CMDCN9
2858  49B0                                    ERR     CMDCOC,CMDCN9
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  49B0  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0032                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49B3  32                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49B4  00                                              .BYTE
 (1)  49B5  CD  15  28        18.0                  CMDCN9::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  49B8  DA  8F  49        10.0                          JC      CMDCOC          ;LOOP ADDRESS IF LOOP SPECIFIED
2859                                        ;>'FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2860
2861  49BB  3E  81               7.0  CMDCOD: MVI     A,$81           ;LOAD MTA REGISTER 1 SELECT CODE
```

```
2862  49BD  D3  40          10.0          OUT     TCMD
2863  49BF  3E  07           7.0          MVI     A,$07           ;ISSUE THE TEST COMMAND
2864  49C1  D3  40          10.0          OUT     TCMD
2865  49C3  3E  80           7.0          MVI     A,$80           ;LOAD MTA REGISTER 0 SELECT CODE
2866  49C5  D3  40          10.0          OUT     TCMD
2867  49C7  DB  40          10.0          IN      TSTS
2868  49C9  E6  40           7.0          ANI     $40             ;READY ON=1
2869  49CB  C2  D3  49      10.0          JNZ     CMDCNB
2870  49CE                                ERR     CMDCOD,CMDCNB
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  49CE  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0033                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49D1  33                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49D2  00                                  .BYTE
 (1)  49D3  CD  15  28      18.0          CMDCNB::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  49D6  DA  BB  49      10.0                  JC      CMDCOD          ;LOOP ADDRESS IF LOOP SPECIFIED
2871                            ;>''RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
2872  49D9  3E  82           7.0          MVI     A,$82           ;LOAD MTA REGISTER 2 SELECT CODE
2873  49DB  D3  40          10.0          OUT     TCMD
2874  49DD  DB  40          10.0          IN      TSTS
2875  49DF  E6  80           7.0          ANI     $80
2876  49E1  C2  E9  49      10.0          JNZ     CMDCNA
2877  49E4                                ERR     CMDCOD,CMDCNA
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  49E4  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0034                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49E7  34                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49E8  00                                  .BYTE
 (1)  49E9  CD  15  28      18.0          CMDCNA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  49EC  DA  BB  49      10.0                  JC      CMDCOD          ;LOOP ADDRESS IF LOOP SPECIFIED
2878                            ;>''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
2879
2880  49EF  3E  82           7.0  CMDCOE: MVI     A,$82           ;LOAD MTA REGISTER 2 SELECT CODE
2881  49F1  D3  40          10.0          OUT     TCMD
2882  49F3  DB  40          10.0          IN      TSTS            ;TU78 BIT=1?
2883  49F5  E6  10           7.0          ANI     $10
2884  49F7  C2  FF  49      10.0          JNZ     CMDCNG          ;YES-CONTINUE
2885  49FA                                ERR     CMDCOE,CMDCNG
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  49FA  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0035                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49FD  35                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49FE  00                                  .BYTE
 (1)  49FF  CD  15  28      18.0          CMDCNG::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4A02  DA  EF  49      10.0                  JC      CMDCOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2886                            ;>''TU78'' IN MTA REGISTER 2 NOT SET
2887
2888  4A05  CD  70  48      18.0  OFFOUT: CALL    CLEAR
2889  4A08  C9              10.0          RET                     ;RETURN TO IN-LINE
2890
```

```
                                        .SBTTL  PROGRAM VARIABLES

2892
2893
2894  4A09    00                DATA:   .BYTE   0               ;DATA FOR THE THRESHOLD TEST
2895  4A0A    00                UNITMP: .BYTE   0               ;UNIT MAP
2896          0000                      .END
```

```
A      =%0007          ADATA  = 0094          ALP01  464D          ALP02  46AD
ALP03  470D            ALP04  476D          ALP05  466E  G       ALP06  46CE  G
ALP07  472E  G         ALP08  478E  G       AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE  4F98          ATTCD  4F97          AXNUM  4F91
B      =%0000          BADST  = 0090        BIT0   = 0001        BIT1   = 0002
BIT15  = 8000          BIT2   = 0004        BIT3   = 0008        BIT4   = 0010
BIT5   = 0020          BIT6   = 0040        BIT7   = 0080        BIT8   = 0100
BRKXCT= 4F00           BRKPBC= 4F0A         BRKRAM= 4F10         BRKSTR= 4E60
BYTEL  4F23            BSAVE  4F9C          BYTCNT= 00D4         BYTEH  4F24
CASSTA= 00A0           C      =%0001        CASCT  4F21          CASCTL= 00A0
CBYTH  = 008B          CATTH  = 0089        CATTL  = 0088        CBUSST= 00A1
CDG2H  = 0093          CBYTL  = 008A        CDG1H  = 0087        CDG1L  = 0086
CDVTH  = 008D          CDG2L  = 0092        CDG3H  = 0095        CDG3L  = 0094
CH1TIE= 0021           CDVTL  = 008C        CHPTIE= 0028         CHOTIE= 0020
CH5TIE= 0025           CH2TIE= 0022         CH3TIE= 0023         CH4TIE= 0024
CLEAR  4870            CH6TIE= 0026         CH7TIE= 0027         CKLOP  = 2815
CMCOH  = 0099          CLKCTL= 00F0         CLOCK  4F26          CLRLP  = 2874
CMC2H  = 009D          CMCOL  = 0098        CMC1H  = 0098        CMC1L  = 009A
CMDCM  48A6            CMC2L  = 009C        CMC3H  = 009F        CMC3L  = 009E
CMDCNZ 48D5  G         CMDCNA 49E9  G       CMDCNB 49D3  G       CMDCNG 49FF  G
CMDCN3 4921  G         CMDCN0 48B9  G       CMDCN1 48F5  G       CMDCN2 4908  G
CMDCN7 4989  G         CMDCN4 493F  G       CMDCN5 4955  G       CMDCN6 4973  G
CMDCOB 495B            CMDCN8 49A3  G       CMDCN9 49B5  G       CMDCOA 4927
CMDCOM 489F            CMDCOC 498F          CMDCOD 49BB          CMDCOE 49EF
CRCWRD= 0018           CMINH  = 0097        CMINL  = 0096        CNTCTL= 00D7
CTCH   = 0085          CSAVE  4F9D          CSRLH  = 0091        CSRLL  = 0090
CXCTH  = 0081          CTCL   = 0084        CTSTH  = 008F        CTSTL  = 008E
C.     = 0001          CXCTL  = 0080        CXINH  = 0083        CXINL  = 0082
C.DTU  = 0003          C.AVAI= 0080         C.DP   = 0008        C.DSE  = 0010
C.FNCT= 003E           C.DVA  = 0008        C.FAIL= 00FC         C.FMT  = 0070
C.NSA  = 0080          C.GO   = 0001        C.INTC= 00FE         C.MAIN= 0020
C.SKPC= 000F           C.RCT  = 00FC        C.SER  = 0080        C.SHR  = 0040
DATA   4A09            C.TAPE= 0040         C.WCS  = 0002        D      =%0002
DBUSST= 00C0           DATACT= 00D0         DBUS   4F28          DBUSCT= 00C0
DDRBIN= 0002           DDRA   = 00D8        DDRAIN= 0010         DDRB   = 00D9
DDRCTL= 00DB           DDRC   = 00DA        DDRCIN= 0001         DDRCO  = 0088
DIARD  = 0008          DIAFLG 4F22          DIAGPG= 4300         DIAGRM= 4F90
DSE    = 0006          DONE1  = 0045        DONINT= 0010         DSAVE  4F9E
D.EOTD= 0010           DUMMY  431E  G       D.ATH0= 0001         D.ATH1= 0002
D.TACH= 0008           D.LAGC= 0020         D.NOTW= 0040         D.NTHR= 0004
ECCCOR= 0019           D.WR4  = 0080        E      =%0003        ECCBAD= 0042
EDATA  = 0095          ECCOK  = 0041        ECCSTA= 001A         ECCTST= 000E
ERLPA  = 280F          EOTCLR= 0003         ERFLG  4F93          ERLP   = 2809
ERRCNT= 00D6           ERLPB  = 2812        ERLPE  = 280C        ERNUM  4F90
E.AMT  = 0020          ESAVE  4F9F          EXIT   486D          E.ACRC= 0010
E.RPE  = 0040          E.CDP  = 0080        E.CRC  = 0080        E.PNTR= 0008
FIFORD= 006A           E.STEC= 0001         E.TTEC= 0002         E.UNC  = 0004
GCRSET= 0002           FORMAT 4F25          FWDTST= 0061         GCRID  = 0089
IE     = 0008          GOODTM= 0092         H      =%0004        HLSAVE 4FA0
I.RMPE= 0040           INTSTA= 00E0         ITERA  4F9A          I.PWR  = 0020
KCALL  = 005F          I5.5   = 0010        I6.5   = 0020        I7.5   = 0040
KEXAM  = 003E          KCLR   = 007B        KDEP   = 003F        KENAB  = 0078
KEY11  = 006E          KEYBRD= 00C8         KEY1   = 0078        KEY10  = 006D
                       KEY12  = 006F        KEY13  = 005C        KEY14  = 005D
```

```
KEY15 = 005E    KEY16 = 005F    KEY17 = 003C    KEY18 = 003D
KEY19 = 003E    KEY2  = 0079    KEY20 = 003F    KEY3  = 007A
KEY4  = 007B    KEY5  = 0074    KEY6  = 0075    KEY7  = 0076
KEY8  = 0077    KEY9  = 006C    KINTA = 006F    KLDAD = 003D
KN0   = 003C    KN1   = 005C    KN2   = 005D    KN3   = 005E
KN4   = 006C    KN5   = 006D    KN6   = 006E    KN7   = 0074
KN8   = 0075    KN9   = 0076    KU2   = 0079    KU3   = 007A
KU8   = 0077    L     =%0005    LBLANK= 000F    LCE   = 0008
LCH   = 000C    LCL   = 000D    LCP   = 000E    LC0   = 0000
LC1   = 0001    LC2   = 0002    LC3   = 0003    LC4   = 0004
LC5   = 0005    LC6   = 0006    LC7   = 0007    LC8   = 0008
LC9   = 0009    LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC
LDLEDD= 00CD    LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800
LKKBD = 004C    LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055
LKLWPG= 0052    LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046
LPFLG   4F94    LPNUM   4F92    M     =%0006    MBSEL = 00E0
MB.A  = 0008    MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A
MH    = 8000    MSF   = 0008    MSGN  = 0035    MTACLR= 0000
MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040
MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080
MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002
MT.REV= 0020    MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080
M.CAPE= 0020    M.CONT= 0080    M.DEM = 0020    M.EBL = 0010
M.EXC = 0008    M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010
M.OCC = 0020    M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080
M.RDEN= 0002    M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001
M.TRA = 0040    M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080
M.WREN= 0080    M5.5  = 0001    M6.5  = 0002    M7.5  = 0004
NOTCAP= 0088    OFFOUT  4A05    OKAY  = 00FF    OPRRAM= 4300
OPSTRT= 0058    OPVER = 0040    PADCNT= 00D5    PADCRC= 0080
PDIAG = 0048    PEID  = 008A    PENAB = 004C    PESET = 0001
PL    = 0081    PRDD  = 004C    PRENF = 009C    PS    = 00B2
PSTAT = 0048    PSW   =%0009    P.AMTP= 0001    P.BCTC= 0040
P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040    P.LWR = 0020
P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002    P.RPOE= 0020
P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080
P.STAT= 0002    P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010
P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001    P.WPEN= 0010
P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004
P.5VOK= 0002    QUE   = 281B    QUEM  = 281E    RAMT  = 0010
RARA  = 0006    RARAI = 0004    RCHBD0= 0048    RCHBD1= 0047
RCHOK = 0046    RCHTST= 000C    RCLRT = 000D    RCMD  = 0008
RCMLP = 0003    RCONT = 0080    RDATA = 0017    RDCLK = 0010
RDON  = 0011    READG = 0007    REND  = 0014    REQST = 2806
RESCHR= 00D1    REVTST= 0064    REWIND= 0004    RFIFOL= 0008
RGCLK = 0002    RGCRI = 0003    RIBG  = 0001    RILL  = 0012
RINST = 000C    RMCTST= 0008    RMK2  = 0013    RNOP  = 0000
RPATH = 0001    RPBAD = 0044    RPCHI = 0001    RPCLK = 0003
RPCTL = 0009    RPEI  = 0002    RPFAIL= 0000    RPF1  = 009D
RPF2  = 009E    RPOK  = 0043    RPOSTN= 0016    RPSTA = 0015
RRCMT = 000A    RSTAT = 0002    RTIEB = 000A    RTIER = 0030
RTM   = 0005    RUNKI = 0009    RUPTST= 005E    RWDUNL= 0005
R.AMT = 0001    R.BOP = 0008    R.DATA= 0040    R.DON = 0002
R.DRDY= 0010    R.END = 0010    R.ILL = 0004    R.JVOK= 0004
```

```
R.MK2  = 0008          R.PLOD= 0008          R.PL00= 0010          R.PL01= 0020
R.POST= 0020          R.STNM= 0002          R.STOP= 0004          R.STPC= 0001
R.TBJN= 0080          R.TSTD= 0040          R.VOK  = 0080          R00H   = 0081
R00L   = 0080          R01H   = 0083          R01L   = 0082          R02H   = 0085
R02L   = 0084          R03H   = 0087          R03L   = 0086          R04H   = 0089
R04L   = 0088          R05H   = 0088          R05L   = 008A          R06H   = 008D
R06L   = 008C          R07H   = 008F          R07L   = 008E          R10H   = 0091
R10L   = 0090          R11H   = 0093          R11L   = 0092          R12H   = 0095
R12L   = 0094          R13H   = 0097          R13L   = 0096          R14H   = 0099
R14L   = 0098          R15H   = 0098          R15L   = 009A          R16H   = 009D
R16L   = 009C          R17H   = 009F          R17L   = 009E          R7.5   = 0010
SELCLR= 0000          SETATA= 00A1          SID    = 0080          SOD    = 0080
SOE    = 0040          SP     =%0008          SSCLK  = 0040          SSTEP  = 0005
STACK  = 4FFF          STATRM= 4F20          STPCT    4F20          STRSP  = 5000
TADR00= 0080          TADR01= 0081          TADR02= 0082          TADR03= 0083
TADR04= 0084          TADR05= 0085          TADR06= 0086          TADR07= 0087
TADR10= 0088          TADR11= 0089          TADR12= 008A          TADR13= 0083
TAMT   = 0044          TASEL  = 0080          TCMD   = 0040          TC.FWD= 0040
TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010
TEMP     4F99          TEST01   4300          TEST02   43E6          TEST03   44AA
TEST04   456E          TEST05   4632          TEST06   4690          TEST07   46F0
TEST10   4750          TEST11   47B0          TEST12   47DD          TEST13   480C
TEST14   483B          TEST15   486A          TMF    = 0099          TMRDY  = 0040
TRKENA= 00D2          TSET   = 2803          TSTEND= 2818          TSTS   = 0040
TST01X   4327          TST02X   43EB          TST03X   44AF          TST04X   4573
TST05X   4637          TST06X   4695          TST07X   46F5          TST10X   4755
TST11X   47BD          TST12X   47EA          TST13X   4819          TST14X   4848
TUSEL0= 00D1          TUSEL1= 00D2          TU78   = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT  = 0004          T.EOT  = 0002          T.FPT  = 0001
T.NTHR= 0004          T.ONL  = 0020          T.PES  = 0008          T.PSBJ= 0020
T.PSOJ= 0008          T.PS1J= 0010          T.RDY  = 0080          T.RDYO= 0040
T.RWD  = 0010          T.SCLK= 0002          T1CN1    4363  G       T1CN2    4386  G
T1CN3    43A9  G       T1CN4    43CC  G       T2CN1    4427  G       T2CN2    444A  G
T2CN3    446D  G       T2CN4    4490  G       T3CN1    44EB  G       T3CN2    450E  G
T3CN3    4531  G       T3CN4    4554  G       T4CN1    45AF  G       T4CN2    45D2  G
T4CN3    45F5  G       T4CN4    4618  G       UIBG   = 00A1          UNITMP   4A0A
VALFC    4F98          VALTB    4F95          VELTST= 005B          WDR.P  = 0010
WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000
WRTDAT= 00D3          W.ACRC= 0004          W.CRC  = 0008          W.DIAG= 0002
W.DONN= 0040          W.ECC  = 0010          W.ENAB= 0080          W.ERR  = 0020
W.FMT  = 0070          W.GCR  = 0010          W.LEFT= 0004          W.ONES= 0020
W.RESI= 0002          W.REV  = 0004          W.ROME= 0010          W.RST  = 0001
W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020          X      =%000A
X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001
X.WCLK= 0001          Y      =%0008          .      = 4A08

ERRORS DETECTED:  0

*MTA2.A78/PTP,MTA2=NLIST,PARAM,MACRO,LIST,MTA2
RUN-TIME: 5 8 0 SECONDS
CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:02 SEQ 0390
PARAM.M80 TABLE OF CONTENTS

```
                                        .TITLE  MTA3 - MAG TAPE ADAPTER TEST PART #3
                                        ;ID MTA3-MAG TAPE ADAPTER CONTROLLER PART #3
                                        .SBTTL  TEST 01 - TACH LINE CONTINUITY TEST - MTA #0
4300                                    ST
                                        ; ************************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*TACH LINE CONTINUITY TEST - MTA #0
4300                                    SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                        ;*THE TACH VALUE READ FROM THE TU PORT.
4300                                    SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*  IF TU PORT #0 NOT SELECTED BY USER
                                        ;*  :  THEN-NEXT TEST
                                        ;*  :  ELSE-CONTINUE
                                        ;*  ENDIF
                                        ;*  CALL SUBROUTINE CLEAR
                                        ;*  SELECT TU PORT #0
                                        ;*  CALL SUBROUTINE TACOM
                                        ;*ENDTST
4300                                    SE
                                        ; ************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*MTA3 MICRO TEST 01
                                        ;*MTA3 MICRO ERROR 01
                                        ;*MTA TACH LINE CONTINUITY TEST - MTA #0
                                        ;*TU CABLE
                                        ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*MTA3 MICRO TEST 01
                                        ;*MTA3 MICRO ERROR 02
                                        ;*MTA TACH LINE CONTINUITY TEST - MTA #0
                                        ;*TU CABLE
                                        ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                        ;*ACTUAL DATA = TUP TACH BIT VALUE
                                        ;*EXPECTED DATA = MTA TACH BIT VALUE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
                                        ; ************************************************************************
4300                                    TEST01: TESTX   @01             ;INITIALIZE THE TEST
4300  3E  01              7.0                       MVI     A,@01       ;DEFINE THE TEST NUMBER
4302  CD  03  28         18.0                       CALL    TSET        ;SETUP THE TEST
                                        ;%MTA TACH LINE CONTINUITY TEST - MTA #0
                                        ;&TU CABLE
```

```
1368  4305                                        REQ     @7,0,0,0,0
 (1)  4305  CD  06  28        18.0                       CALL    REQST
 (1)  4308  00                                           .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                       .WORD   0               ;SYSTEM '0' COUNT
 (1)  430B  00  00                                       .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                           .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                           .BYTE   @7              ;REQUEST CODE
1369  430F                                        RIN     R12L                    ;GET THE UNITS DESIRED
 (1)  430F  DB  94            10.0                       IN      R12L                    ;READ R12L INTO AC
 (1)  4311  7F                 4.0                       MOV     A,A                     ;RETRY LINK
1370  4312  32  31  44        13.0                STA     UNITMP                  ;STORE IN MEMORY
1371  4315  A7                 4.0                ANA     A                       ;SET THE CONDITION CODE
1372  4316  C2  2F  43        10.0                JNZ     TST01X                  ;GO TEST THE SPECIFIED UNITS
1373                                                                            ;NO UNITS SPECIFIED
1374  4319                                        ERR     EXIT,DUMMY
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28        18.0                       CALL    ERLP                    ;PROCESS ERROR - DO 2.3
 (1)        0001                                         MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                           .BYTE
 (1)  431E  CD  15  28        18.0                DUMMY:: CALL    CKLOP                   ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  DC  43        10.0                        JC      EXIT                    ;LOOP ADDRESS IF LOOP SPECIFIED
1375                                            ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1376                                            ;>FATAL ERROR - TEST ABORTED
1377  4324  C3  DC  43        10.0                JMP     EXIT                    ;EXIT THE TEST
1378  4327  3A  31  44        13.0                LDA     UNITMP                  ;GET THE UNIT MAP
1379  432A  E6  01             7.0                ANI     @001                    ;TEST UNIT 0?
1380  432C  CA  4F  43        10.0                JZ      TEST02                  ;NO-GO CHECK TEST 2
1381  432F  CD  DF  43        18.0        TST01X: CALL    CLEAR
1382  4332  DB  E0             10.0                IN      INTSTA
1383  4334  E6  80              7.0                ANI     BIT7
1384  4336  D3  E0             10.0                OUT     MBSEL
1385  4338  CD  0A  44        18.0                CALL    TACOM
1386  433B                                        ENDTST  TST01X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  433B                                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  433B  CD  06  28        18.0                       CALL    REQST
 (2)  433E  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  433F  00  00                                       .WORD                   ;SYSTEM '0' COUNT
 (2)  4341  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4343  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4344  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4345  3A  9A  4F        13.0                LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4348  3D                 4.0                DCR     A                       ;DOWNCOUNT
 (1)  4349  32  9A  4F        13.0                STA     ITERA                   ;SAVE COUNT
 (1)  434C  F2  2F  43        10.0                JP      TST01X                  ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 02 - TACH LINE CONTINUITY TEST - MTA #1
                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*TACH LINE CONTINUITY TEST - MTA #1
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                ;*THE TACH VALUE READ FROM THE TU PORT.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #1 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #1
                                ;*  CALL SUBROUTINE TACOM
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA3 MICRO TEST 02
                                ;*MTA3 MICRO ERROR 02
                                ;*MTA TACH LINE CONTINUITY TEST - MTA #1
                                ;*TU CABLE
                                ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                ;*ACTUAL DATA = TUP TACH BIT VALUE
                                ;*EXPECTED DATA = MTA TACH BIT VALUE
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                S
                                ; ***************************************************************************
4351 CD 03 28        18.0       TEST02: TESTX   @02              ;INITIALIZE THE TEST
4351 CD 03 28        18.0                       MVI     A,@02     ;DEFINE THE TEST NUMBER
4351 CD 03 28        18.0                       CALL    TSFT      ;SETUP THE TEST
                                ;%MTA TACH LINE CONTINUITY TEST - MTA #1
                                ;&TU CABLE
4354 3A 31 44        13.0               LDA     UNITMP           ;GET THE UNIT MAP
4357 E6 02            7.0               ANI     @002             ;TEST UNIT 1?
4359 CA 7E 43        10.0               JZ      TEST03           ;NO-GO CHECK TEST 3
435C CD DF 43        18.0       TST02X: CALL    CLEAR
435F DB E0           10.0               IN      INTSTA
4361 E6 80            7.0               ANI     BIT7
4363 F6 01            7.0               ORI     $01
4365 D3 E0           10.0               OUT     MBSEL
4367 CD 0A 44        18.0               CALL    TACOM
```

```
1427  436A                                        ENDTST  TST02X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  436A                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  436A  CD  06  28        18.0                                CALL    REQST
 (2)  436D  00                                                    .BYTE                   ;DATA PATTERN NUMBER
 (2)  436E  00  00                                                .WORD                   ;SYSTEM '*' COUNT
 (2)  4370  00  00                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4372  00                                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4373  07                                                    .BYTE   7               ;REQUEST CODE
 (1)  4374  3A  9A  4F        13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  4377  3D                 4.0           DCR     A               ;DOWNCOUNT
 (1)  4378  32  9A  4F        13.0           STA     ITERA           ;SAVE COUNT
 (1)  437B  F2  5C  43        10.0           JP      TST02X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 03 - TACH LINE CONTINUITY TEST - MTA #2
437E                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*TACH LINE CONTINUITY TEST - MTA #2
437E                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                ;*THE TACH VALUE READ FROM THE TU PORT.
437E                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   IF TU PORT #2 NOT SELECTED BY USER
                                ;*   :  THEN-NEXT TEST
                                ;*   :  ELSE-CONTINUE
                                ;*   ENDIF
                                ;*   CALL SUBROUTINE CLEAR
                                ;*   SELECT TU PORT #2
                                ;*   CALL SUBROUTINE TACOM
                                ;*ENDTST
437E                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA3 MICRO TEST 03
                                ;*MTA3 MICRO ERROR 02
                                ;*MTA TACH LINE CONTINUITY TEST - MTA #2
                                ;*TU CABLE
                                ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                ;*ACTUAL DATA = TUP TACH BIT VALUE
                                ;*EXPECTED DATA = MTA TACH BIT VALUE
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
437E                            S
                                ;*********************************************************************
437E                            TEST03: TESTX   @03             ;INITIALIZE THE TEST
437E  3E 03          7.0                        MVI     A,@03           ;DEFINE THE TEST NUMBER
4380  CD 03 28      18.0                        CALL    TSET            ;SETUP THE TEST
                                ;%MTA TACH LINE CONTINUITY TEST - MTA #2
                                ;&TU CABLE
4383  3A 31 44      13.0              LDA     UNITMP          ;GET THE UNIT MAP
4386  E6 04          7.0              ANI     @004            ;TEST UNIT 2?
4388  CA AD 43      10.0              JZ      TEST04          ;NO-GO CHECK TEST 4
438B  CD DF 43      18.0  TST03X:     CALL    CLEAR
438E  DB E0         10.0              IN      INTSTA
4390  E6 80          7.0              ANI     BIT7
4392  F6 02          7.0              ORI     $02
4394  D3 E0         10.0              OUT     MBSEL
4396  CD 0A 44      18.0              CALL    TACOM
```

```
1468  4399                                          ENDTST  TST03X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4399                                         REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4399  CD  06  28        18.0                                 CALL    REQST
 (2)  439C  00                                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  439D  00  00                                                 .WORD                   ;SYSTEM '#' COUNT
 (2)  439F  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43A1  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43A2  07                                                     .BYTE   7               ;REQUEST CODE
 (1)  43A3  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  43A6  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  43A7  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  43AA  F2  8B  43        10.0          JP      TST03X          ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 04 - TACH LINE CONTINUITY TEST - MTA #3
                                        ST
                                        ;*********************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*TACH LINE CONTINUITY TEST - MTA #3
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                        ;*THE TACH VALUE READ FROM THE TU PORT.
                                        SP
                                        ;*********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*  IF TU PORT #3 NOT SELECTED BY USER
                                        ;*  :  THEN-NEXT TEST
                                        ;*  :  ELSE-CONTINUE
                                        ;*  ENDIF
                                        ;*  CALL SUBROUTINE CLEAR
                                        ;*  SELECT TU PORT #3
                                        ;*  CALL SUBROUTINE TACOM
                                        ;*ENDTST
                                        SE
                                        ;*********************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*MTA3 MICRO TEST 04
                                        ;*MTA3 MICRO ERROR 02
                                        ;*MTA TACH LINE CONTINUITY TEST - MTA #3
                                        ;*TU CABLE
                                        ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                        ;*ACTUAL DATA = TUP TACH BIT VALUE
                                        ;*EXPECTED DATA = MTA TACH BIT VALUE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        S
                                        ;*********************************************************************
43AD                                    TEST04: TESTX   @04             ;INITIALIZE THE TEST
 (1) 43AD  3E  04          7.0                          MVI     A,@04     ;DEFINE THE TEST NUMBER
 (1) 43AF  CD  03  28     18.0                          CALL    TSET      ;SETUP THE TEST
                                        ;%MTA TACH LINE CONTINUITY TEST - MTA #3
                                        ;&TU CABLE
43B2  3A  31  44          13.0                  LDA     UNITMP          ;GET THE UNIT MAP
43B5  E6  08               7.0                  ANI     @010            ;TEST UNIT 3?
43B7  CA  DC  43          10.0                  JZ      EXIT            ;NO-GO CHECK TEST 5
43BA  CD  DF  43          18.0          TST04X: CALL    CLEAR
43BD  DB  E0              10.0                  IN      INTSTA
43BF  E6  80               7.0                  ANI     BIT7
43C1  F6  03               7.0                  ORI     $03
43C3  D3  E0              10.0                  OUT     MBSEL
43C5  CD  0A  44          18.0                  CALL    TACOM
```

```
1509  43C8                                          ENDTST  TST04X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43C8                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43C8  CD  06  28        18.0                                  CALL    REQST
 (2)  43CB  00                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  43CC  00  00                                                  .WORD                   ;SYSTEM '=' COUNT
 (2)  43CE  00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43D0  00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43D1  07                                                      .BYTE   7               ;REQUEST CODE
 (1)  43D2  3A  9A  4F        13.0              LDA     ITERA               ;GET ITERATION COUNT
 (1)  43D5  3D                 4.0              DCR     A                   ;DOWNCOUNT
 (1)  43D6  32  9A  4F        13.0              STA     ITERA               ;SAVE COUNT
 (1)  43D9  F2  BA  43        10.0              JP      TST04X              ;DO TEST UNTIL TILL = 0
1510
1511
1512  43DC  C3  18  28        10.0  EXIT:       JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************
1531  43DF  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
1532  43E0  C5              12.0          PUSH    B
1533  43E1  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
1534  43E3  DB  E0           10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
1535  43E5  E6  80            7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
1536  43E7  B0                4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1537  43E8  D3  E0           10.0          OUT     MBSEL           ;RESET TO THIS PORT
1538  43EA  3E  80            7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
1539  43EC  D3  40           10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
1540  43EE  /.                4.0          XRA     A               ;CLEAR TU COMMAND A
1541  43EF  D3  40           10.0          OUT     TCMD            ;
1542  43F1  3E  81            7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
1543  43F3  D3  40           10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
1544  43F5  3E  00            7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
1545  43F7  D3  40           10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
1546  43F9  AF                4.0          XRA     A
1547  43FA  D3  44           10.0          OUT     TAMT            ;CLEAR AMTIE WORD
1548  43FC  D3  48           10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
1549  43FE  D3  4C           10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
1550  4400  04                4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
1551  4401  78                4.0          MOV     A,B
1552  4402  FE  04            7.0          CPI     4               ;DONE?
1553  4404  C2  E3  43       10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
1554  4407  C1               10.0          POP     B               ;RESET B & C
1555  4408  F1               10.0          POP     PSW             ;ALL DONE
1556  4409  C9               10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE  ACH COMMON
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*TACH CONTINUITY COMMON CODE
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE CHECKS THE VALUE OF THE TACH BIT ON THE MTA BOARD AND
                                        ;*EXPECTES THE TACH BIT ON THE TUP BOARD TO MATCH.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*  SELECT MTA REGISTER 5
                                        ;*  INPUT THE MTA TACH BIT
                                        ;*  INPUT THE TUP TACH BIT
                                        ;*  IF MTA TACH BIT=TUP TACH BIT
                                        ;*  :  THEN-CONTINUE
                                        ;*  :  ELSE-ERROR
                                        ;*  ENDIF
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************
1575 440A 3E 85            7.0  TACOM:  MVI     A,$85          ;LOAD MTA REGISTER 5 SELECT CODE
1576 440C D3 40           10.0         OUT     TCMD
1577 440E DB 40           10.0         IN      TSTS           ;INPUT THE TU DIAG REGISTER
1578 4410 E6 08            7.0         ANI     BIT3           ;SAVE THE MTA TACH BIT
1579 4412 1F               4.0         RAR
1580 4413 1F               4.0         RAR
1581 4414 1F               4.0         RAR
1582 4415 47               4.0         MOV     B,A            ;SAVE IN B
1583 4416 D3 95           10.0         OUT     EDATA          ;EDATA=MTA TACH BIT
1584 4418 DB 48           10.0         IN      PSTAT          ;INPUT THE TU PORT TACH BIT
1585 441A E6 08            7.0         ANI     BIT3
1586 441C 1F               4.0         RAR
1587 441D 1F               4.0         RAR
1588 441E 1F               4.0         RAR
1589 441F B8               4.0         CMP     B              ;BOTH THE SAME?
1590 4420 CA 2A 44        10.0         JZ      TACON          ;YES-CONTINUE
1591 4423 D3 94           10.0         OUT     ADATA          ;ADATA=TUP TACH BIT
```

```
1593  4425                                          ERRB    TACOM,TACON
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4425  CD  12  28        18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0002                                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4428  02                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4429  00                                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  442A  CD  15  28        18.0                    TACON:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  442D  DA  0A  44        10.0                            JC      TACOM           ;LOOP ADDRESS IF LOOP SPECIFIED
1594                                                ;>MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
1595                                                ;>ACTUAL DATA = TUP TACH BIT VALUE
1596                                                ;>EXPECTED DATA = MTA TACH BIT VALUE
1597  4430  C9                    10.0                    RET
1598
1599
1600
1601                                                .SBTTL  PROGRAM VARIABLES
1602  4431  00                                      UNITMP: .BYTE   0
1603        0000                                            .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F98           ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLEAR   43DF           CLKCTL= 00F0           CLOCK   4F26          CLRLP   43E3
CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAVE   4F9D           CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082          C.     = 0001
C.AVAI= 0080           C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002          D      =%0002          DATACT= 00D0
DBUS    4F28           DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300           DIAGRM= 4F90           DIARD  = 000B          DONE1  = 0045
DONINT= 0010           DSAVE   4F9E           DSE    = 0006          DUMMY   431E   G
D.ATH0= 0001           D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020
D.NOTW= 0040           D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E           EDATA  = 0095         EOTCLR= 0003
ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812
ERLPE  = 280C          ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F
EXIT    43DC           E.ACRC= 0010           E.AMT  = 0020         E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040         E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25
FWDTST= 0061           GCRID  = 0089          GCRSET= 0002          GOODTM= 0092
H      =%0004          HLSAVE  4FA0           IE     = 0008         INTSTA= 00E0
ITERA   4F9A           I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F         KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E         KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D          KEY11  = 006E         KEY12  = 006F
KEY13  = 005C          KEY14  = 005D          KEY15  = 005E         KEY16  = 005F
KEY17  = 003C          KEY18  = 003D          KEY19  = 003E         KEY2   = 0079
KEY20  = 003F          KEY3   = 007A          KEY4   = 007B         KEY5   = 0074
KEY6   = 0075          KEY7   = 0076          KEY8   = 0077         KEY9   = 006C
KINTA  = 006F          KLDAD  = 003D          KN0    = 003C         KN1    = 005C
KN2    = 005D          KN3    = 005E          KN4    = 006C         KN5    = 006D
KN6    = 006E          KN7    = 0074          KN8    = 0075         KN9    = 0076
KU2    = 0079          KU3    = 007A          KU8    = 0077         L      =%0005
```

```
LBLANK= 000F    LCE   = 0008    LCH   = 000C    LCL   = 000D
LCP   = 000E    LC0   = 0000    LC1   = 0001    LC2   = 0002
LC3   = 0003    LC4   = 0004    LC5   = 0005    LC6   = 0006
LC7   = 0007    LC8   = 0008    LC9   = 0009    LDLEDA= 00CA
LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD    LDLEDE= 00CE
LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C    LKKEY = 0049
LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052    LKLWPP= 004F
LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94    LPNUM   4F92
M     =%0006    MBSEL = 00E0    MB.A  = 0008    MB.B  = 0004
MEMTOP= 4FFF    MINUS = 000A    MM    = 8000    MSE   = 0008
MSGN  = 0002    MTACLR= 0000    MT.ARA= 0020    MT.CPE= 0080
MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004
MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004
MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010
MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020    M.CONT= 0080
M.DEM = 0020    M.EBL = 0010    M.EXC = 0008    M.FAIL= 0008
M.ILR = 0010    M.INIT= 0010    M.OCC = 0020    M.ONLI= 0001
M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008
M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040    M.UNIT= 0007
M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080    M5.5  = 0001
M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088    OKAY  = 00FF
OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040    PADCNT= 00D5
PADCRC= 0080    PDIAG = 0048    PEID  = 008A    PENAB = 004C
PESET = 0001    PL    = 0081    PRDD  = 004C    PRENF = 009C
PS    = 00B2    PSTAT = 0048    PSW   =%0009    P.AMTP= 0001
P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040
P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002
P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010
P.SING= 0080    P.STAT= 0002    P.STPE= 0080    P.TACH= 0008
P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001
P.WPEN= 0010    P.WP0E= 0008    P.WP1E= 0004    P.WP2E= 0008
P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B    QUEM  = 281E
RAMT  = 0010    RARA  = 0006    RARAI = 0004    RCHBD0= 0048
RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C    RCLRT = 000D
RCMD  = 0008    RCMLP = 0003    RCONT = 0080    RDATA = 0017
RDCLK = 0010    RDON  = 0011    READG = 0007    REND  = 0014
REQST = 2806    RESCHR= 00D1    REVTST= 0064    REWIND= 0004
RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003    RIBG  = 0001
RILL  = 0012    RINST = 000C    RMCTST= 0008    RMK2  = 0013
RNOP  = 0000    RPATH = 0001    RPBAD = 0044    RPCHI = 0001
RPCLK = 0003    RPCTL = 0009    RPEI  = 0002    RPFAIL= 0000
RPF1  = 009D    RPF2  = 009E    RPOK  = 0043    RPOSTN= 0016
RPSTA = 0015    RRCMT = 000A    RSTAT = 0002    RTIEB = 000A
RTIER = 0030    RTM   = 0005    RUNKI = 0009    RUPTST= 005E
RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008    R.DATA= 0040
R.DON = 0002    R.DRDY= 0010    R.END = 0010    R.ILL = 0004
R.JVOK= 0004    R.MK2 = 0008    R.PLOD= 0008    R.PLOO= 0010
R.PLO1= 0020    R.POST= 0020    R.STNM= 0002    R.STOP= 0004
R.STPC= 0001    R.TBJN= 0080    R.TSTD= 0040    R.VOK = 0080
R00H  = 0081    R00L  = 0080    R01H  = 0083    R01L  = 0082
R02H  = 0085    R02L  = 0084    R03H  = 0087    R03L  = 0086
R04H  = 0089    R04L  = 0088    R05H  = 008B    R05L  = 008A
R06H  = 008D    R06L  = 008C    R07H  = 008F    R07L  = 008E
R10H  = 0091    R10L  = 0090    R11H  = 0093    R11L  = 0092
```

```
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          TACOM   440A          TACON   442A  G       TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST01  4300          TEST02  434F          TEST03  437E          TEST04  43AD
TMF   = 0099          TMRDY = 0040          TRKENA= 00D2          TSET  = 2803
TSTEND= 2818          TSTS  = 0040          TST01X  432F          TST02X  435C
TST03X  438B          TST04X  438A          TUSEL0= 00D1          TUSEL1= 00D2
TU78  = 0010          T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004
T.EOT = 0002          T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020
T.PES = 0008          T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010
T.RDY = 0080          T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002
UIBG  = 00A1          UNITMP  4431          VALFC   4F98          VALTB   4F95
VELTST= 005B          WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA
WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004
W.CRC = 0008          W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010
W.ENAB= 0080          W.ERR = 0020          W.FMT = 0070          W.GCR = 0010
W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002          W.REV = 0004
W.ROME= 0010          W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008
W.XFER= 0020          X     =%000A          Y.DONN= 0080          X.ENAB= 0040
X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001          Y     =%0008
.     = 4432


ERRORS DETECTED:  0

*MTA3.A78/PTP,MTA3=NLIST,PARAM,MACRO,LIST,MTA3
RUN-TIME:  3 3 0 SECONDS
CORE USED:  70K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:08 SEQ 0405
PARAM.M80 TABLE OF CONTENTS

```
                                        .TITLE  XMC1 - TRANSLATOR MICRO CONTROLLER PART #1
                                        .SBTTL  TEST 1 - XMC - SELF TEST
                                        ;ID     XMC1-TRANSLATOR MICRO CONTROLLER PART #1
1332  4300                              $T
                                        ; ***************************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*XMC - SELF TEST
1334  4300                              $D
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST COMMANDS THE TRANSLATOR MICRO CONTROLLER TO RUN ITS INTERNAL
                                        ;*SELF TEST DIAGNOSTIC PACKAGE.  A SOFTWARE DELAY IS PERFORMED AND THE
                                        ;*XMC ROM PARITY BIT IS TESTED.  IF THE PARITY BIT IS RESET NO SELF TEST
                                        ;*ERROR OCCURED, BUT IF SET EITHER A SELF TEST ERROR WAS DETECTED OR A
                                        ;*ROM PARITY ERROR DETECTED.
1340  4300                              $P
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*  RESET THE MTA IF ONE SELECTED
                                        ;*  SET THE WMC/XMC RESTART BIT
                                        ;*  CLEAR CLOCK CONTROL BYTE
                                        ;*  CLEAR THE WMC/XMC RESTART BIT
                                        ;*  DELAY 1 MS
                                        ;*  IF XMC ROM PARITY ERROR=1
                                        ;*  :  THEN-ERROR
                                        ;*  :  ELSE-CONTINUE
                                        ;*  ENDIF
                                        ;*ENDTST
1352  4300                              $E
                                        ; ***************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*XMC1 MICRO TEST 01
                                        ;*XMC1 MICRO ERROR 01
                                        ;*XMC1-XMC-SELF TEST
                                        ;*M8958, M8959, M8960
                                        ;*XMC ROM PARITY ERROR SET AFTER XMC DISABLE INDICATES XMC SELF
                                        ;*DIAG. FAILURE OR XMC ROM PARITY ERROR.
1359  4300                              $
                                        ; ***************************************************************************

1361  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                       MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                       CALL    TSET            ;SETUP THE TEST
                                        ;%XMC1-XMC-SELF TEST
                                        ;&M8958, M8959, M8960

1365  4305  AF                 4.0              XRA     A               ;CLEAR ALL TRACK ENABLE LINES
1366  4306  D3  D2              10.0              OUT     TRKENA
1367  4308  D3  D2              10.0              OUT     TRKENA
```

```
1368  430A  3E  40        7.0          MVI   A,P.LCS          ;GET RID OF MTA ON THE SYSTEM
1369  430C  D3  48       10.0          OUT   PDIAG
1370  430E  3E  00        7.0          MVI   A,@0             ;SET TO 0
1371  4310  D3  40       10.0          OUT   TCMD             ;CLEAR FOR ANY INTERFEARENCE
1372
1373  4312  3E  01        7.0  XLLP1:  MVI   A,W.RST          ;CLEAR THE XMC ENABLE BIT
1374  4314  D3  D3       10.0          OUT   WMCCTL           ;DISABLE THE XMC
1375  4316  3E  00        7.0          MVI   A,0
1376  4318  D3  F0       10.0          OUT   CLKCTL           ;SET CLOCK CONTROL TO NORMAL OPERATION
1377  431A  AF            4.0          XRA   A
1378  431B  D3  D3       10.0          OUT   WMCCTL           ;FINISH THE RESTART OF WMC/XMC
1379                                                         ;XMC SHOULD START SELF TEST
1380  431D  3E  FF        7.0          MVI   A,@377           ;LOAD A COUNT VALUE
1381  431F  00            4.0  1$:     NOP                    ;NOP
1382  4320  3D            4.0          DCR   A                ;DECREMENT COUNT
1383  4321  C2  1F  43    10.0          JNZ   1$               ;CONTINUE UNTIL COUNT = 0
1384                                                         ;
1385  4324  DB  E0       10.0          IN    INTSTA           ;GET THE INTERRUPT STATUS BYTE
1386  4326  E6  01        7.0          ANI   X.ROME           ;XMC ROM PARITY ERROR?
1387  4328  CA  30  43    10.0          JZ    XLCN1            ;CONTINUE IF CLEAR
1388                                                         ;
1389  432B                               ERR   XLLP1,XLCN1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  432B  CD  09  28    18.0                    CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0001                                  MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  432E  01                                    .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  432F  00                                    .BYTE
 (1)  4330  CD  15  28    18.0          XLCN1:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4333  DA  12  43    10.0                  JC    XLLP1            ;LOOP ADDRESS IF LOOP SPECIFIED
1390                                   ;>XMC ROM PARITY ERROR SET AFTER XMC DISABLE INDICATES XMC SELF
1391                                   ;>DIAG. FAILURE OR XMC ROM PARITY ERROR.
1392  4336                               ENDTST  XLLP1
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4336                             REQ   7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4336  CD  06  28    18.0                  CALL  REQST
 (2)  4339  00                                 .BYTE               ;DATA PATTERN NUMBER
 (2)  433A  00  00                             .WORD               ;SYSTEM '@' COUNT
 (2)  433C  00  00                             .WORD               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  433E  00                                 .BYTE               ;DATA COMPARE FLAG IF =1
 (2)  433F  07                                 .BYTE  7            ;REQUEST CODE
 (1)  4340  3A  9A  4F    13.0          LDA   ITERA            ;GET ITERATION COUNT
 (1)  4343  3D             4.0          DCR   A                ;DOWNCOUNT
 (1)  4344  32  9A  4F    13.0          STA   ITERA            ;SAVE COUNT
 (1)  4347  F2  12  43    10.0          JP    XLLP1            ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - XMC - TRACK ENABLE TEST
1395  434A                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - TRACK ENABLE TEST
1397  434A                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
1398                            ;*THIS TEST LOADS ALL ONES IN THE TRACK ENABLE REGISTER AND COMMANDS THE
1399                            ;*TRANSLATOR MICRO CONTROLLER TO WRITE GCR ALL ONES.  THE TRANSLATOR MICRO
1400                            ;*CONTROLLER IS THEN SINGLE STEPPED UNTIL THE TRANSLATOR MICRO CONTROLLER
1401                            ;*(XMC) STATUS BIT 'XL WCLK' SETS.  THEN THE OUTPUT OF THE TRANSLATOR IS
1402                            ;*CHECKED AT THE TU PORT BOARD FOR THE FIRST DATA STATE OF THE NRZI FORMAT.
1403                            ;*THEN THE XMC IS CLOCKED 8 MORE TIMES AND THE OUTPUT OF THE TRANSLATOR
1404                            ;*IS CHECKED AT THE TU PORT BOARD FOR THE SECOND DATA STATE OF THE NRZI
1405                            ;*FORMAT.
1406                            ;*
1407                            ;*ONCE THE ABOVE SEQUENCE IS COMPLETE THE TEST PROCEEDS TO DISABLE EACH
1408                            ;*OF THE NINE TRACKS ONE AT A TIME UNTIL ALL NINE ARE DISABLED.  AFTER
1409                            ;*EACH TRACK IS DISABLED THE NRZI DATA STATES ARE CHECKED AT THE TU PORT
1410                            ;*BOARD BEFORE THE NEXT TRACK IS DISABLED.
1411  434A                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
1412                            ;*BGNTST
1413                            ;*  REQUEST TU PORTS TO USE FOR TEST
1414                            ;*  SET WRITE PATH ENABLE FOR ALL SELECTED TU PORTS
1415                            ;*  SET THE TRACK ENABLE MASK TO ALL ONES (9 BITS)
1416                            ;*  SET THE PASSCOUNT TO 1
1417                            ;*  SET THE EXPECTED GCR DATA TO ALL ONES (9 BITS)
1418                            ;*  SET THE WMC/XMC RESTAR TBIT
1419                            ;*  SELECT THE SINGLE STEP WRITE CLOCK
1420                            ;*  CLEAR THE WMC/XMC RESTART BIT
1421                            ;*  CLEAR THE TRACK ENABLE REGISTER
1422                            ;*  SET WMC CONTROL BITS X.ENABLE, ONES, GCR AND WRITE
1423                            ;*  BGNDO
1424                            ;*     CALL SUBROUTINE ADJDAT
1425                            ;*     ISSUE THE TRACK ENABLE MASK TO THE TRACK ENABLE REGISTER
1426                            ;*  :  SET THE LOOP COUNT TO 177(8)
1427                            ;*  :  BGNDO
1428                            ;*  :  :  CALL SUBROUTINE CLKSYS
1429                            ;*  :  :  DECREMENT THE LOOP COUNT
1430                            ;*  :  :  DO UNTIL THE LOOP COUNT=0 OR XL WRITE CLOCK SET
1431                            ;*  :  ENDDO
1432                            ;*  :  IF LOOP COUNT=0
1433                            ;*  :  :  THEN-ERROR-TIMEOUT WAITING FOR XL WRITE CLOCK
1434                            ;*  :  :  ELSE-CONTINUE
1435                            ;*  :  ENDIF
1436                            ;*  :  CALL SUBROUTINE CLKSYS
1437                            ;*  :  CALL SUBROUTINE CLKSYS
1438                            ;*  :  IF TU PORT #0 SELECTED BY THE USER
```

```
1439                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #0
1440                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1441                                    ;*  :  :  :  :  THEN-CONTINUE
1442                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1443                                    ;*  :  :  :  ENDIF
1444                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #0
1445                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1446                                    ;*  :  :  :  :  THEN-CONTINUE
1447                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1448                                    ;*  :  :  :  ENDIF
1449                                    ;*  :  :  ELSE-CONTINUE
1450                                    ;*  :  ENDIF
1451                                    ;*  :  IF TU PORT #1 SELECTED BY THE USER
1452                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #1
1453                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1454                                    ;*  :  :  :  :  THEN-CONTINUE
1455                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1456                                    ;*  :  :  :  ENDIF
1457                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #1
1458                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1459                                    ;*  :  :  :  :  THEN-CONTINUE
1460                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1461                                    ;*  :  :  :  ENDIF
1462                                    ;*  :  :  ELSE-CONTINUE
1463                                    ;*  :  ENDIF
1464                                    ;*  :  IF TU PORT #2 SELECTED BY THE USER
1465                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #2
1466                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1467                                    ;*  :  :  :  :  THEN-CONTINUE
1468                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1469                                    ;*  :  :  :  ENDIF
1470                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #2
1471                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1472                                    ;*  :  :  :  :  THEN-CONTINUE
1473                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1474                                    ;*  :  :  :  ENDIF
1475                                    ;*  :  :  ELSE-CONTINUE
1476                                    ;*  :  ENDIF
1477                                    ;*  :  IF TU PORT #3 SELECTED BY THE USER
1478                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #3
1479                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1480                                    ;*  :  :  :  :  THEN-CONTINUE
1481                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1482                                    ;*  :  :  :  ENDIF
1483                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #3
1484                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1485                                    ;*  :  :  :  :  THEN-CONTINUE
1486                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1487                                    ;*  :  :  :  ENDIF
1488                                    ;*  :  :  ELSE-CONTINUE
1489                                    ;*  :  ENDIF
1490                                    ;*  :  CALL SUBROUTINE COMP
1491                                    ;*  :  CALL SUBROUTINE ADJDAT
1492                                    ;*  :  SET THE LOOP COUNT TO 14(8)
```

```
1493                                    ;*  :  BGNDO
1494                                    ;*  :  :  CALL SUBROUTINE CLKSYS
1495                                    ;*  :  :  DECREMENT THE LOOP COUNT
1496                                    ;*  :  :  DO UNTIL THE LOOP COUNT=0
1497                                    ;*  :  ENDDO
1498                                    ;*  :  IF XL WRITE CLOCK=1
1499                                    ;*  :  :  THEN-CONTINUE
1500                                    ;*  :  :  ELSE-ERROR-XL WRITE CLOCK LATE
1501                                    ;*  :  ENDIF
1502                                    ;*  :  CALL SUBROUTINE CLKSYS
1503                                    ;*  :  CALL SUBROUTINE CLKSYS
1504                                    ;*  :  IF TU PORT #0 SELECTED BY THE USER
1505                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #0
1506                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1507                                    ;*  :  :  :  :  THEN-CONTINUE
1508                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1509                                    ;*  :  :  :  ENDIF
1510                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #0
1511                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1512                                    ;*  :  :  :  :  THEN-CONTINUE
1513                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1514                                    ;*  :  :  :  ENDIF
1515                                    ;*  :  :  ELSE-CONTINUE
1516                                    ;*  :  ENDIF
1517                                    ;*  :  IF TU PORT #1 SELECTED BY THE USER
1518                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #1
1519                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1520                                    ;*  :  :  :  :  THEN-CONTINUE
1521                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1522                                    ;*  :  :  :  ENDIF
1523                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #1
1524                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1525                                    ;*  :  :  :  :  THEN-CONTINUE
1526                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1527                                    ;*  :  :  :  ENDIF
1528                                    ;*  :  :  ELSE-CONTINUE
1529                                    ;*  :  ENDIF
1530                                    ;*  :  IF TU PORT #2 SELECTED BY THE USER
1531                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #2
1532                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1533                                    ;*  :  :  :  :  THEN-CONTINUE
1534                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1535                                    ;*  :  :  :  ENDIF
1536                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #2
1537                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1538                                    ;*  :  :  :  :  THEN-CONTINUE
1539                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1540                                    ;*  :  :  :  ENDIF
1541                                    ;*  :  :  ELSE-CONTINUE
1542                                    ;*  :  ENDIF
1543                                    ;*  :  IF TU PORT #3 SELECTED BY THE USER
1544                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #3
1545                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1546                                    ;*  :  :  :  :  THEN-CONTINUE
```

```
1547                                ;*  :  :  :  : ELSE-ERROR-DATA STATE NOT=EXPECTED
1548                                ;*  :  :  :  ENDIF
1549                                ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #3
1550                                ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1551                                ;*  :  :  :  : THEN-CONTINUE
1552                                ;*  :  :  :  : ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1553                                ;*  :  :  :  ENDIF
1554                                ;*  :  :  ELSE-CONTINUE
1555                                ;*  :  ENDIF
1556                                ;*  :  CALL SUBROUTINE COMP
1557                                ;*  :  INCREMENT THE PASS COUNT
1558                                ;*  :  ROTATE THE TRACK ENABLE MASK RIGHT ONE BIT POSITION
1559                                ;*  :  DO UNTIL THE TRACK ENABLE MASK=0
1560                                ;*  ENDDO
1561                                ;*ENDTST
1562  434A                          SE
 (1)                                ; ************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1563                                ;*XMC1 MICRO TEST 02
1564                                ;*XMC1 MICRO ERROR 02
1565                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1566                                ;*M8958, M8959, M8960, M8955'S
1567                                ;*OPERATOR ERROR - NO TM78 UNITS SELECTED
1568                                ;*FATAL ERROR - TEST ABORTED
1569                                ;*
1570                                ;*XMC1 MICRO TEST 02
1571                                ;*XMC1 MICRO ERROR 03
1572                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1573                                ;*M8958, M8959, M8960, M8955'S
1574                                ;*XMC TIMEOUT WHILE WAITING FOR ''XL WCLK'' TO SET
1575                                ;*FATAL ERROR - TEST ABORTED
1576                                ;*
1577                                ;*XMC1 MICRO TEST 02
1578                                ;*XMC1 MICRO ERROR 04
1579                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1580                                ;*M8958, M8959, M8960, M8955'S
1581                                ;*FIRST GCR DATA STATE NOT AS EXPECTED
1582                                ;*WHEN READ FROM TUO STATUS BYTE
1583                                ;*ACTUAL = NNNN
1584                                ;*EXPECTED = NNNN
1585                                ;*
1586                                ;*XMC1 MICRO TEST 02
1587                                ;*XMC1 MICRO ERROR 05
1588                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1589                                ;*M8958, M8959, M8960, M8955'S
1590                                ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1591                                ;*WHEN READ FROM ''WCSP'' BIT OF PORT 0 STATUS BYTE
1592                                ;*ACTUAL = NNNN
1593                                ;*EXPECTED = NNNN
1594                                ;*
1595                                ;*XMC1 MICRO TEST 02
1596                                ;*XMC1 MICRO ERROR 06
1597                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
```

```
1598                    ;*M8958, M8959, M8960, M8955'S
1599                    ;*FIRST GCR DATA STATE NOT AS EXPECTED
1600                    ;*WHEN READ FROM TU1 STATUS BYTE
1601                    ;*ACTUAL = NNNN
1602                    ;*EXPECTED = NNNN
1603                    ;*
1604                    ;*XMC1 MICRO TEST 02
1605                    ;*XMC1 MICRO ERROR 07
1606                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1607                    ;*M8958, M8959, M8960, M8955'S
1608                    ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1609                    ;*WHEN READ FROM 'WCSP'' BIT OF PORT 1 STATUS BYTE
1610                    ;*ACTUAL = NNNN
1611                    ;*EXPECTED = NNNN
1612                    ;*
1613                    ;*XMC1 MICRO TEST 02
1614                    ;*XMC1 MICRO ERROR 10
1615                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1616                    ;*M8958, M8959, M8960, M8955'S
1617                    ;*FIRST GCR DATA STATE NOT AS EXPECTED
1618                    ;*WHEN READ FROM TU2 STATUS BYTE
1619                    ;*ACTUAL = NNNN
1620                    ;*EXPECTED = NNNN
1621                    ;*
1622                    ;*XMC1 MICRO TEST 02
1623                    ;*XMC1 MICRO ERROR 11
1624                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1625                    ;*M8958, M8959, M8960, M8955'S
1626                    ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1627                    ;*WHEN READ FROM 'WCSP'' BIT OF PORT 2 STATUS BYTE
1628                    ;*ACTUAL = NNNN
1629                    ;*EXPECTED = NNNN
1630                    ;*
1631                    ;*XMC1 MICRO TEST 02
1632                    ;*XMC1 MICRO ERROR 12
1633                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1634                    ;*M8958, M8959, M8960, M8955'S
1635                    ;*FIRST GCR DATA STATE NOT AS EXPECTED
1636                    ;*WHEN READ FROM TU3 STATUS BYTE
1637                    ;*ACTUAL = NNNN
1638                    ;*EXPECTED = NNNN
1639                    ;*
1640                    ;*XMC1 MICRO TEST 02
1641                    ;*XMC1 MICRO ERROR 13
1642                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1643                    ;*M8958, M8959, M8960, M8955'S
1644                    ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1645                    ;*WHEN READ FROM 'WCSP'' BIT OF PORT 3 STATUS BYTE
1646                    ;*ACTUAL = NNNN
1647                    ;*EXPECTED = NNNN
1648                    ;*
1649                    ;*XMC1 MICRO TEST 02
1650                    ;*XMC1 MICRO ERROR 14
1651                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
```

```
1652                    ;*M8958, M8959, M8960, M8955'S
1653                    ;*XMC 'XL WCLK'' NOT SET 8 CLOCKS AFTER FIRST DATA STATE
1654                    ;*
1655                    ;*XMC1 MICRO TEST 02
1656                    ;*XMC1 MICRO ERROR 15
1657                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1658                    ;*M8958, M8959, M8960, M8955'S
1659                    ;*SECOND GCR DATA STATE NOT AS EXPECTED
1660                    ;*WHEN READ FROM TU0 STATUS BYTE
1661                    ;*ACTUAL = NNNN
1662                    ;*EXPECTED = NNNN
1663                    ;*
1664                    ;*XMC1 MICRO TEST 02
1665                    ;*XMC1 MICRO ERROR 16
1666                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1667                    ;*M8958, M8959, M8960, M8955'S
1668                    ;*SECOND GCR PARITY STATE NOT AS EXPECTED
1669                    ;*WHEN READ FROM 'WCSP'' BIT OF PORT 0 STATUS BYTE
1670                    ;*ACTUAL = NNNN
1671                    ;*EXPECTED = NNNN
1672                    ;*
1673                    ;*XMC1 MICRO TEST 02
1674                    ;*XMC1 MICRO ERROR 17
1675                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1676                    ;*M8958, M8959, M8960, M8955'S
1677                    ;*SECOND GCR DATA STATE NOT AS EXPECTED
1678                    ;*WHEN READ FROM TU1 STATUS BYTE
1679                    ;*ACTUAL = NNNN
1680                    ;*EXPECTED = NNNN
1681                    ;*
1682                    ;*XMC1 MICRO TEST 02
1683                    ;*XMC1 MICRO ERROR 20
1684                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1685                    ;*M8958, M8959, M8960, M8955'S
1686                    ;*SECOND GCR PARITY STATE NOT AS EXPECTED
1687                    ;*WHEN READ FROM 'WCSP'' BIT OF PORT 1 STATUS BYTE
1688                    ;*ACTUAL = NNNN
1689                    ;*EXPECTED = NNNN
1690                    ;*
1691                    ;*XMC1 MICRO TEST 02
1692                    ;*XMC1 MICRO ERROR 21
1693                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1694                    ;*M8958, M8959, M8960, M8955'S
1695                    ;*SECOND GCR DATA STATE NOT AS EXPECTED
1696                    ;*WHEN READ FROM TU2 STATUS BYTE
1697                    ;*ACTUAL = NNNN
1698                    ;*EXPECTED = NNNN
1699                    ;*
1700                    ;*XMC1 MICRO TEST 02
1701                    ;*XMC1 MICRO ERROR 22
1702                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1703                    ;*M8958, M8959, M8960, M8955'S
1704                    ;*SECOND GCR PARITY STATE NOT AS EXPECTED
1705                    ;*WHEN READ FROM 'WCSP'' BIT OF PORT 2 STATUS BYTE
```

```
1706                                    ;*ACTUAL = NNNN
1707                                    ;*EXPECTED = NNNN
1708                                    ;*
1709                                    ;*XMC1 MICRO TEST 02
1710                                    ;*XMC1 MICRO ERROR 23
1711                                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1712                                    ;*M8958, M8959, M8960, M8955'S
1713                                    ;*SECOND GCR DATA STATE NOT AS EXPECTED
1714                                    ;*WHEN READ FROM TU3 STATUS BYTE
1715                                    ;*ACTUAL = NNNN
1716                                    ;*EXPECTED = NNNN
1717                                    ;*
1718                                    ;*XMC1 MICRO TEST 02
1719                                    ;*XMC1 MICRO ERROR 24
1720                                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1721                                    ;*M8958, M8959, M8960, M8955'S
1722                                    ;*SECOND GCR PARITY STATE NOT AS EXPECTED
1723                                    ;*WHEN READ FROM 'WCSP'' BIT OF PORT 3 STATUS BYTE
1724                                    ;*ACTUAL = NNNN
1725                                    ;*EXPECTED = NNNN
1726  434A                              S
 (1)                                    ; ****************************************************************************
1727
1728  434A                              TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  434A  3E  02            7.0                MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  434C  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1729                                    ;%XMC1-XMC TRACK ENABLE TEST-GCR
1730                                    ;&M8958, M8959, M8960, M8955'S
1731  434E                                      REQ     @7,0,0,0,0
 1)  434F  CD  06  28       18.0                CALL    REQST
 1)  4352  00                                   .BYTE   0               ;DATA PATTERN NUMBER
 1)  4353  00  00                               .WORD   0               ;SYSTEM "0" COUNT
 1)  4355  00  00                               .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 1)  4357  00                                   .BYTE   0               ;DATA COMPARE FLAG IF =1
 1)  4358  07                                   .BYTE   @7              ;REQUEST CODE
1732  4359                                      RIN     R12L
 (1)  4359  DB  94           10.0               IN      R12L            ;READ R12L INTO AC
 (1)  435B  7F                4.0               MOV     A,A             ;RETRY LINK
1733  435C  32  76  47        13.0               STA     UNITMP
1734
1735  435F  A7                 4.0               ANA     A
1736  4360  C2  71  43        10.0               JNZ     UNIT0
1737  4363                                      ERR     TSTEND,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4363  CD  09  28       18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4366  02                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4367  00                               .BYTE
 (1)  4368  CD  15  28       18.0            DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  436B  DA  18  28       10.0                JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1738                                    ;>OPERATOR ERROR - NO TM78 UNITS SELECTED
1739                                    ;<FATAL ERROR - TEST ABORTED
1740  436E  C3  18  28       10.0               JMP     TSTEND
```

```
1742
1743                                    ;
1744                                    ;TAPE UNITS PRESENT ROUTINE - DATA PATH ENABLE AND LOOP WRITE TO READ
1745                                    ;WILL BE SET FOR ALL UNITS PRESENT.
1746                                    ;
1747
1748  4371  CD 4A 47      18.0  UNIT0:  CALL   CLEAR
1749  4374  3A 76 47      13.0          LDA    UNITMP          ;GET THE UNIT MAP BYTE
1750  4377  E6 01          7.0          ANI    $01             ;UNIT 0 PRESENT?
1751  4379  CA 86 43      10.0          JZ     UNIT1           ;NO-CONTINUE
1752  437C  DB E0         10.0          IN     INTSTA          ;
1753  437E  E6 80          7.0          ANI    BIT7            ;
1754  4380  D3 E0         10.0          OUT    MBSEL           ;
1755  4382  3E 10          7.0          MVI    A,P.WPEN
1756  4384  D3 4C         10.0          OUT    PENAB           ;CLEAR TAPE CMD WORD
1757
1758  4386  3A 76 47      13.0  UNIT1:  LDA    UNITMP          ;GET THE UNIT MAP BYTE
1759  4389  E6 02          7.0          ANI    $02             ;UNIT 1 PRESENT?
1760  438B  CA 9A 43      10.0          JZ     UNIT2           ;NO-CONTINUE
1761  438E  DB E0         10.0          IN     INTSTA          ;
1762  4390  E6 80          7.0          ANI    BIT7            ;
1763  4392  F6 01          7.0          ORI    @01             ;
1764  4394  D3 E0         10.0          OUT    MBSEL           ;
1765  4396  3E 10          7.0          MVI    A,P.WPEN
1766  4398  D3 4C         10.0          OUT    PENAB           ;CLEAR TAPE CMD WORD
1767
1768  439A  3A 76 47      13.0  UNIT2:  LDA    UNITMP          ;GET THE UNIT MAP BYTE
1769  439D  E6 04          7.0          ANI    $04             ;UNIT 2 PRESENT?
1770  439F  CA AE 43      10.0          JZ     UNIT3           ;NO-CONTINUE
1771  43A2  DB E0         10.0          IN     INTSTA          ;
1772  43A4  E6 80          7.0          ANI    BIT7            ;
1773  43A6  F6 02          7.0          ORI    @02             ;
1774  43A8  D3 E0         10.0          OUT    MBSEL           ;
1775  43AA  3E 10          7.0          MVI    A,P.WPEN
1776  43AC  D3 4C         10.0          OUT    PENAB
1777
1778  43AE  3A 76 47      13.0  UNIT3:  LDA    UNITMP          ;GET THE UNIT MAP BYTE
1779  43B1  E6 08          7.0          ANI    $08             ;UNIT 3 PRESENT?
1780  43B3  CA C2 43      10.0          JZ     GO              ;NO-START TEST
1781  43B6  DB E0         10.0          IN     INTSTA          ;
1782  43B8  E6 80          7.0          ANI    BIT7            ;
1783  43BA  F6 03          7.0          ORI    @03             ;
1784  43BC  D3 E0         10.0          OUT    MBSEL           ;
1785  43BE  3E 10          7.0          MVI    A,P.WPEN
1786  43C0  D3 4C         10.0          OUT    PENAB
1787
1788
1789                                    ;
1790                                    ;ALL AVAILABLE PORTS ARE SET UP - START THE TEST SEQUENCE
1791                                    ;
1792
1793  43C2  3E 40          7.0  GO:     MVI    A,P.LCS         ;KILL MTA ON THIS PORT
1794  43C4  D3 48         10.0          OUT    PDIAG
1795  43C6  AF             4.0          XRA    A
```

```
1796  43C7  D3  40      10.0          OUT   TCMD              ;KILL MTA LINK
1797  43C9  3E  01       7.0          MVI   A,$1              ;LOAD A 1
1798  43CB  32  7A 47   13.0          STA   EGCRP             ;LOAD THE EXPECTED PARITY STATE
1799  43CE  32  78 47   13.0          STA   TEPMSK            ;LOAD THE INITIAL TRACK ENABLE MASK
1800  43D1  32  75 47   13.0          STA   PASSCNT           ;LOAD UP THE PASS COUNT
1801  43D4  3E  FF       7.0          MVI   A,$FF
1802                                                         ;SET THE DATA TRACK ENABLE
1803  43D6  32  77 47   13.0          STA   TEDMSK            ;MASK TO ALL ONES
1804  43D9  32  7B 47   13.0          STA   EGCRD             ;SET THE EXPECTED DATA STATE
1805  43DC  3E  01       7.0          MVI   A,W.RST           ;RESTART THE WMC/XMC
1806  43DE  D3  D3      10.0          OUT   WMCCTL            ;LOAD THE CONTROL
1807  43E0  3E  05       7.0          MVI   A,SSTEP           ;SELECT SINGLE STEP WRITE CLOCK
1808  43E2  D3  F0      10.0          OUT   CLKCTL            ;INITIALIZE CLOCK MODULE
1809  43E4  32  7C 47   13.0          STA   CCTLWD            ;SAVE SOFTWARE IMAGE
1810  43E7  AF           4.0          XRA   A
1811  43E8  D3  D3      10.0          OUT   WMCCTL            ;FINISH THE RESTART CYCLE
1812  43EA  AF           4.0          XRA   A                 ;CLEAR THE ACCUMULATOR
1813  43EB  D3  D2      10.0          OUT   TRKENA            ;CLEAR THE TRACK ENABLE REGISTER
1814  43ED  D3  D2      10.0          OUT   TRKENA
1815
1816  43EF  3E  78       7.0          MVI   A,X.ENAB!W.ONES!W.GCR!W.WRITE
1817  43F1  D3  D3      10.0          OUT   WMCCTL            ;START THE XMC
1818
1819  43F3  CD  35 47   18.0  ENLP0:  CALL  ADJDAT            ;ADJUST THE EXPECTED DATA
1820  43F6  3A  75 47   13.0          LDA   PASSCNT           ;GET THE PASS COUNT
1821  43F9                           ROUT  R05L              ;WRITE TO CAS
 (1)  43F9  D3  8A      10.0                OUT   R05L                ;WRITE AC INTO R05L
 (1)  43FB  7F           4.0                MOV   A,A                 ;RETRY LINK
1822  43FC  AF           4.0          XRA   A                 ;CLEAR A
1823  43FD                           ROUT  R05H              ;WRITE TO CAS
 (1)  43FD  D3  8B      10.0                OUT   R05H                ;WRITE AC INTO R05H
 (1)  43FF  7F           4.0                MOV   A,A                 ;RETRY LINK
1824  4400  3A  78 47   13.0          LDA   TEPMSK            ;GET THE TRACK ENABLE PARITY BIT
1825  4403  D3  D2      10.0          OUT   TRKENA            ;LOAD TO THE TRACK ENABLE BYTE
1826  4405  3A  77 47   13.0          LDA   TEDMSK            ;GET THE TRACK ENABLE DATA BITS
1827  4408  D3  D2      10.0          OUT   TRKENA            ;LOAD TO THE TRACK ENABLE BYTE
1828
1829  440A  3E  7F       7.0          MVI   A,@177            ;LOAD A TIMEOUT COUNT
1830  440C  32  79 47   13.0          STA   TOUT
1831
1832  440F  CD  18 47   18.0  1$:     CALL  CLKSYS            ;SINGLE STEP THE XMC
1833
1834  4412  3A  79 47   13.0          LDA   TOUT              ;GET THE TIMEOUT COUNT
1835  4415  3D           4.0          DCR   A                 ;DECREMENT COUNT
1836  4416  32  79 47   13.0          STA   TOUT              ;STORE THE TIMEOUT COUNT
1837  4419  DB  D0      10.0          IN    WMCSTA            ;GET THE WMC STATUS BYTE
1838  441B  E6  01       7.0          ANI   X.WCLK            ;XL WCLK SET?
1839  441D  C2  35 44   10.0          JNZ   TUNITO            ;YES TO CHECK DATA
1840  4420  3A  79 47   13.0          LDA   TOUT              ;GET THE TIMEOUT COUNT
1841  4423  A7           4.0          ANA   A                 ;SET CONDITION BITS
1842  4424  C2  0F 44   10.0          JNZ   1$
```

```
                                        ERR     UNIT0,ENCN0
                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 4427 CD 09 28  18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)      0003                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 442A 03                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 442B 00                                     .BYTE
(1) 442C CD 15 28  18.0                 ENCN0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1) 442F DA 71 43  10.0                         JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>XMC TIMEOUT WHILE WAITING FOR 'XL WCLK'' TO SET
                                ;>FATAL ERROR - TEST ABORTED

4432 C3 18 28      10.0                 JMP     TSTEND

                                ;
                                ;TAPE UNIT PORT 0 TEST - DATA PHASE 0
                                ;
4435 CD 18 47      18.0         TUNIT0: CALL    CLKSYS          ;ISSUE ONE MORE CLOCK TO THE XMC
4438 CD 18 47      18.0                 CALL    CLKSYS
443B 21 7B 47      10.0                 LXI     H,EGCRD
443E 3A 76 47      13.0                 LDA     UNITMP          ;GET THE UNIT MAP BYTE
4441 E6 01          7.0                 ANI     $01             ;UNIT 0 PRESENT?
4443 CA 89 44      10.0                 JZ      TUNIT1          ;NO-CONTINUE
4446 DB E0         10.0                 IN      INTSTA          ;
4448 E6 80          7.0                 ANI     BIT7            ;
444A D3 E0         10.0                 OUT     MBSEL           ;
444C DB 40         10.0                 IN      TSTS            ;YES-GET THE TU STATUS
444E 32 7D 47      13.0                 STA     TEMPD
4451 BE             7.0                 CMP     M               ;COMPARE WITH EXPECTED DATA
4452 CA 61 44      10.0                 JZ      TU0CN0          ;CONTINUE IF EQUAL
4455                                    ROUT    ADATA           ;STORE ACTUAL DATA
(1) 4455 D3 94     10.0                         OUT     ADATA           ;WRITE AC INTO ADATA
(1) 4457 7F         4.0                         MOV     A,A             ;RETRY LINK
4458 7E             7.0                 MOV     A,M             ;GET EXPECTED DATA
4459                                    ROUT    EDATA           ;STORE EXPECTED DATA
(1) 4459 D3 95     10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
(1) 445B 7F         4.0                         MOV     A,A             ;RETRY LINK
445C                                    ERRB    UNIT0,TU0CN0
(1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
(1) 445C CD 12 28  18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
(1)      0004                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 445F 04                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 4460 00                                     .BYTE                   ;PRINT ROUTINE NUMBER
(1) 4461 CD 15 28  18.0                 TU0CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1) 4464 DA 71 43  10.0                         JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>FIRST GCR DATA STATE NOT AS EXPECTED
                                ;>WHEN READ FROM TU0 STATUS BYTE

4467 21 7A 47      10.0                 LXI     H,EGCRP
446A DB 48         10.0                 IN      PSTAT
446C 0F             4.0                 RRC
446D 0F             4.0                 RRC
446E E6 01          7.0                 ANI     $01
4470 32 7E 47      13.0                 STA     TEMPP
4473 BE             7.0                 CMP     M
4474 CA 83 44      10.0                 JZ      TU0CN1
```

```
1880  4477                                          ROUT    ADATA
 (1)  4477  D3  94            10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4479  7F                 4.0                          MOV     A,A             ;RETRY LINK
1881  447A  7E                 7.0                  MOV     A,M
1882  447B                                          ROUT    EDATA
 (1)  447B  D3  95            10.0                          OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  447D  7F                 4.0                          MOV     A,A             ;RETRY LINK
1883  447E                                          ERRB    UNIT0,TU0CN1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  447E  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4481  05                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4482  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4483  CD  15  28        18.0                  TU0CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4486  DA  71  43        10.0                          JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
1884                                          ;>FIRST GCR PARITY STATE NOT AS EXPECTED
1885                                          ;>WHEN READ FROM 'WCSP'' BIT OF PORT 0 STATUS BYTE
1886
1887  4489                                          S
 (1)                                              ; ***************************************************************************
1888                                          ;TAPE UNIT PORT 1 TEST - DATA PHASE 0
1889  4489                                          S
 (1)                                              ; ***************************************************************************
1890
1891  4489  21  7B  47        10.0          TUNIT1: LXI     H,EGCRD
1892  448C  3A  76  47        13.0                  LDA     UNITMP          ;GET THE UNIT MAP BYTE
1893  448F  E6  02             7.0                  ANI     $02             ;UNIT 1 PRESENT?
1894  4491  CA  D9  44        10.0                  JZ      TUNIT2          ;NO-CONTINUE
1895  4494  DB  E0            10.0                  IN      INTSTA          ;
1896  4496  E6  80             7.0                  ANI     BIT7            ;
1897  4498  F6  01             7.0                  ORI     @01             ;
1898  449A  D3  E0            10.0                  OUT     MBSEL           ;
1899  449C  DB  40            10.0                  IN      TSTS            ;YES-GET THE TU STATUS
1900  449E  32  7D  47        13.0                  STA     TEMPD
1901  44A1  BE                 7.0                  CMP     M               ;COMPARE WITH EXPECTED DATA
1902  44A2  CA  B1  44        10.0                  JZ      TU1CN0          ;CONTINUE IF EQUAL
1903  44A5                                          ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  44A5  D3  94            10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44A7  7F                 4.0                          MOV     A,A             ;RETRY LINK
1904  44A8  7E                 7.0                  MOV     A,M             ;GET EXPECTED DATA
1905  44A9                                          ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  44A9  D3  95            10.0                          OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44AB  7F                 4.0                          MOV     A,A             ;RETRY LINK
1906  44AC                                          ERRB    UNIT0,TU1CN0
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44AC  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44AF  06                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B0  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44B1  CD  15  28        18.0                  TU1CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44B4  DA  71  43        10.0                          JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
1907                                          ;>FIRST GCR DATA STATE NOT AS EXPECTED
1908                                          ;>WHEN READ FROM TU1 STATUS BYTE
1909
```

```
1910  44B7  21  7A  47      10.0          LXI     H,EGCRP
1911  44BA  DB  48          10.0          IN      PSTAT
1912  44BC  0F               4.0          RRC
1913  44BD  0F               4.0          RRC
1914  44BE  E6  01           7.0          ANI     $01
1915  44C0  32  7E  47      13.0          STA     TEMPP
1916  44C3  BE               7.0          CMP     M
1917  44C4  CA  D3  44      10.0          JZ      TU1CN1
1918  44C7                                ROUT    ADATA
 (1)  44C7  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44C9  7F               4.0                  MOV     A,A             ;RETRY LINK
1919  44CA  7E               7.0          MOV     A,M
1920  44CB                                ROUT    EDATA
 (1)  44CB  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44CD  7F               4.0                  MOV     A,A             ;RETRY LINK
1921  44CE                                ERRB    UNIT0,TU1CN1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44CE  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D1  07                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D2  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44D3  CD  15  28      18.0          TU1CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D6  DA  71  43      10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
1922                                ;>FIRST GCR PARITY STATE NOT AS EXPECTED
1923                                ;>WHEN READ FROM 'WCSP'' BIT OF PORT 1 STATUS BYTE
1924
1925  44D9                          S
 (1)                                ; **********************************************************************
1926                                ;TAPE UNIT PORT 2 TEST - DATA PHASE 0
1927  44D9                          S
 (1)                                ; **********************************************************************
1928
1929  44D9  21  7B  47      10.0    TUNIT2: LXI     H,EGCRD
1930  44DC  3A  76  47      13.0            LDA     UNITMP          ;GET THE UNIT MAP BYTE
1931  44DF  E6  04           7.0            ANI     $04             ;UNIT 2 PRESENT?
1932  44E1  CA  29  45      10.0            JZ      TUNIT3          ;NO-CONTINUE
1933  44E4  DB  E0          10.0            IN      INTSTA          ;
1934  44E6  E6  80           7.0            ANI     BIT7            ;
1935  44E8  F6  02           7.0            ORI     @02             ;
1936  44EA  D3  E0          10.0            OUT     MBSEL           ;
1937  44EC  DB  40          10.0            IN      TSTS            ;YES-GET THE TU STATUS
1938  44EE  32  7D  47      13.0            STA     TEMPD
1939  44F1  BE               7.0            CMP     M               ;COMPARE WITH EXPECTED DATA
1940  44F2  CA  01  45      10.0            JZ      TU2CN0          ;CONTINUE IF EQUAL
1941  44F5                                  ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  44F5  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44F7  7F               4.0                  MOV     A,A             ;RETRY LINK
1942  44F8  7E               7.0            MOV     A,M             ;GET EXPECTED DATA
1943  44F9                                  ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  44F9  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44FB  7F               4.0                  MOV     A,A             ;RETRY LINK
```

```
                                        ERRB    UNIT0,TU2CN0
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
                                                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                      .BYTE                   ;PRINT ROUTINE NUMBER
                                              TU2CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                      JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>FIRST GCR DATA STATE NOT AS EXPECTED
                                ;>WHEN READ FROM TJ2 STATUS BYTE

                                        LXI     H,EGCRP
                                        IN      PSTAT
                                        RRC
                                        RRC
                                        ANI     $01
                                        STA     TEMPP
                                        CMP     M
                                        JZ      TU2CN1
                                        ROUT    ADATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MOV     A,M
                                        ROUT    EDATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    UNIT0,TU2CN1
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
                                                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                      .BYTE                   ;PRINT ROUTINE NUMBER
                                              TU2CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                      JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>FIRST GCR PARITY STATE NOT AS EXPECTED
                                ;>WHEN READ FROM 'WCSP'' BIT OF PORT 2 STATUS BYTE


                                S
                                ; ***************************************************************************************
                                ;TAPE UNIT PORT 3 TEST - DATA PHASE 0
                                S
                                ; ***************************************************************************************

                                TUNIT3: LXI     H,EGCRD
                                        LDA     UNITMP          ;GET THE UNIT MAP BYTE
                                        ANI     $08             ;UNIT 3 PRESENT?
                                        JZ      PHASE2          ;NO-CONTINUE
                                        IN      INTSTA
                                        ANI     BIT7
                                        ORI     @03
                                        OUT     MBSEL
                                        IN      TSTS            ;YES-GET THE TU STATUS
                                        STA     TEMPD
                                        CMP     M               ;COMPARE WITH EXPECTED DATA
```

```
 1979  4542  CA  51  45      10.0          JZ      TU3CN0              ;CONTINUE IF EQUAL
 1980  4545                                ROUT    ADATA               ;STORE ACTUAL DATA
  (1)  4545  D3  94          10.0                   OUT     ADATA               ;WRITE AC INTO ADATA
  (1)  4547  7F               4.0                   MOV     A,A                 ;RETRY LINK
 1981  4548  7E               7.0          MOV     A,M                 ;GET EXPECTED DATA
 1982  4549                                ROUT    EDATA               ;STORE EXPECTED DATA
  (1)  4549  D3  95          10.0                   OUT     EDATA               ;WRITE AC INTO EDATA
  (1)  454B  7F               4.0                   MOV     A,A                 ;RETRY LINK
 1983  454C                                ERRB    UNITO,TU3CN0
  (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  454C  CD  12  28      18.0                   CALL    ERLPB               ;PROCESS ERROR - DO 2.3
  (1)        000A                                 MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  454F  0A                                   .BYTE   MSGN                ;MESSAGE NUMBER ID
  (1)  4550  00                                   .BYTE                       ;PRINT ROUTINE NUMBER
  (1)  4551  CD  15  28      18.0          TU3CN0::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4554  DA  71  43      10.0                   JC      UNITO               ;LOOP ADDRESS IF LOOP SPECIFIED
 1984                                   ;>FIRST GCR DATA STATE NOT AS EXPECTED
 1985                                   ;>WHEN READ FROM TU3 STATUS BYTE
 1986  4557  21  7A  47      10.0          LXI     H,EGCRP
 1987  455A  DB  48          10.0          IN      PSTAT
 1988  455C  0F               4.0          RRC
 1989  455D  0F               4.0          RRC
 1990  455E  E6  01           7.0          ANI     $01
 1991  4560  32  7E  47      13.0          STA     TEMPP
 1992  4563  BE               7.0          CMP     M
 1993  4564  CA  73  45      10.0          JZ      TU3CN1
 1994  4567                                ROUT    ADATA
  (1)  4567  D3  94          10.0                   OUT     ADATA               ;WRITE AC INTO ADATA
  (1)  4569  7F               4.0                   MOV     A,A                 ;RETRY LINK
 1995  456A  7E               7.0          MOV     A,M
 1996  456B                                ROUT    EDATA
  (1)  456B  D3  95          10.0                   OUT     EDATA               ;WRITE AC INTO EDATA
  (1)  456D  7F               4.0                   MOV     A,A                 ;RETRY LINK
 1997  456E                                ERRB    UNITO,TU3CN1
  (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  456E  CD  12  28      18.0                   CALL    ERLPB               ;PROCESS ERROR - DO 2.3
  (1)        0008                                 MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4571  08                                   .BYTE   MSGN                ;MESSAGE NUMBER ID
  (1)  4572  00                                   .BYTE                       ;PRINT ROUTINE NUMBER
  (1)  4573  CD  15  28      18.0          TU3CN1::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4576  DA  71  43      10.0                   JC      UNITO               ;LOOP ADDRESS IF LOOP SPECIFIED
 1998                                   ;>FIRST GCR PARITY STATE NOT AS EXPECTED
 1999                                   ;>WHEN READ FROM 'WCSP'' BIT OF PORT 3 STATUS BYTE
 2000
 2001  4579  CD  24  47      18.0  PHASE2: CALL    COMP                ;COMPLEMENT THE EXPECTED DATA
 2002  457C  CD  35  47      18.0          CALL    ADJDAT              ;ADJUST THE EXPECTED DATA
 2003  457F  06  0E           7.0  2$:     MVI     B,14                ;SET UP THE CLOCK COUNT
 2004  4581  CD  18  47      18.0  3$:     CALL    CLKSYS              ;CLOCK THE TRANSLATOR
 2005  4584  05               4.0          DCR     B                   ;DECREMENT THE CLOCK COUNT
 2006  4585  C2  81  45      10.0          JNZ     3$                  ;CONTINUE-TILL ZERO
 2007  4588  DB  D0          10.0          IN      WMCSTA              ;GET THE WMC STATUS
 2008  458A  E6  01           7.0          ANI     $01
 2009  458C  C2  94  45      10.0          JNZ     PH2CN
```

```
2011  458F                                          ERR     UNIT0,PH2CN
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  458F  CD  09  28      18.0                              CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4592  0C                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4593  00                                                .BYTE
 (1)  4594  CD  15  28      18.0                    PH2CN:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4597  DA  71  43      10.0                            JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2012                                        ;>XMC ''XL WCLK'' NOT SET 8 CLOCKS AFTER FIRST DATA STATF
2013
2014
2015                                        ;
2016                                        ;TAPE UNIT PORT 0 TEST - DATA PHASE 1
2017                                        ;
2018
2019  459A  CD  18  47      18.0  EUNIT0: CALL    CLKSYS          ;ISSUE ONE MORE CLOCK TO THE XMC
2020  459D  CD  18  47      18.0          CALL    CLKSYS
2021  45A0  21  7B  47      10.0          LXI     H,EGCRD
2022  45A3  3A  76  47      13.0          LDA     UNITMP          ;GET THE UNIT MAP BYTE
2023  45A6  E6  01           7.0          ANI     $01             ;UNIT 0 PRESENT?
2024  45A8  CA  EE  45      10.0          JZ      EUNIT1          ;NO-CONTINUE
2025  45AB  DB  E0          10.0          IN      INTSTA
2026  45AD  E6  80           7.0          ANI     BIT7
2027  45AF  D3  E0          10.0          OUT     MBSEL
2028  45B1  DB  40          10.0          IN      TSTS            ;YES-GET THE TU STATUS
2029  45B3  32  7D  47      13.0          STA     TEMPD
2030  45B6  BE               7.0          CMP     M               ;COMPARE WITH EXPECTED DATA
2031  45B7  CA  C6  45      10.0          JZ      EUOCN0          ;CONTINUE IF EQUAL
2032  45BA                                ROUT    ADATA           ;STORE ACEUAL DATA
 (1)  45BA  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45BC  7F               4.0                  MOV     A,A             ;RETRY LINK
2033  45BD  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
2034  45BE                                ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  45BE  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45C0  7F               4.0                  MOV     A,A             ;RETRY LINK
2035  45C1                                ERRB    UNIT0,EUOCN0
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  45C1  CD  12  28      18.0                              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000D                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45C4  0D                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45C5  00                                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45C6  CD  15  28      18.0                    EUOCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45C9  DA  71  43      10.0                            JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2036                                        ;>SECOND GCR DATA STATE NOT AS EXPECTED
2037                                        ;>WHEN READ FROM TUO STATUS BYTE
2038
2039
2040  45CC  21  7A  47      10.0          LXI     H,EGCRP
2041  45CF  DB  48          10.0          IN      PSTAT
2042  45D1  0F               4.0          RRC
2043  45D2  0F               4.0          RRC
2044  45D3  E6  01           7.0          ANI     $01
2045  45D5  32  7E  47      13.0          STA     TEMPP
2046  45D8  BE               7.0          CMP     M
```

```
2047  45D9  CA  E8  45   10.0          JZ      EU0CN1
2048  45DC                             ROUT    ADATA
 (1)  45DC  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45DE  7F             4.0                  MOV     A,A             ;RETRY LINK
2049  45DF  7E             7.0          MOV     A,M
2050  45E0                             ROUT    EDATA
 (1)  45E0  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45E2  7F             4.0                  MOV     A,A             ;RETRY LINK
2051  45E3                             ERRB    UNIT0,EU0CN1
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  45E3  CD  12  28    18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000E                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45E6  0E                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45E7  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45E8  CD  15  28    18.0          EU0CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45EB  DA  71  43    10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2052                             ;>SECOND GCR PARITY STATE NOT AS EXPECTED
2053                             ;>WHEN READ FROM 'WCSP'' BIT OF PORT 0 STATUS BYTE
2054
2055  45EE                       S
 (1)                             ; ****************************************.****************************************
2056                             ;TAPE UNIT PORT 1 TEST - DATA PHASE ONE
2057  45EE                       S
 (1)                             ; ********************************************************************************
2058
2059  45EE  21  7B  47    10.0  EUNIT1: LXI     H,EGCRD
2060  45F1  3A  76  47    13.0          LDA     UNITMP          ;GET THE UNIT MAP BYTE
2061  45F4  E6  02         7.0          ANI     $02             ;UNIT 1 PRESENT?
2062  45F6  CA  3E  46    10.0          JZ      EUNIT2          ;NO-CONTINUE
2063  45F9  DB  E0        10.0          IN      INTSTA
2064  45FB  E6  80         7.0          ANI     BIT7
2065  45FD  F6  01         7.0          ORI     @01
2066  45FF  D3  E0        10.0          OUT     MBSEL
2067  4601  DB  40        10.0          IN      TSTS            ;YES-GET THE TU STAEUS
2068  4603  32  7D  47    13.0          STA     TEMPD
2069  4606  BE             7.0          CMP     M               ;COMPARE WITH EXPECTED DATA
2070  4607  CA  16  46    10.0          JZ      EU1CN0          ;CONTINUE IF EQUAL
2071  460A                             ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  460A  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  460C  7F             4.0                  MOV     A,A             ;RETRY LINK
2072  460D  7E             7.0          MOV     A,M             ;GET EXPECTED DATA
2073  460E                             ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  460E  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4610  7F             4.0                  MOV     A,A             ;RETRY LINK
2074  4611                             ERRB    UNIT0,EU1CN0
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4611  CD  12  28    18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4614  0F                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4615  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4616  CD  15  28    18.0          EU1CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4619  DA  71  43    10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2075                             ;>SECOND GCR DATA STATE NOT AS EXPECTED
2076                             ;>WHEN READ FROM TU1 STATUS BYTE
```

```
2077
2078  461C  21  7A  47   10.0          LXI     H,EGCRP
2079  461F  DB  48        10.0          IN      PSTAT
2080  4621  0F           4.0           RRC
2081  4622  0F           4.0           RRC
2082  4623  E6  01       7.0           ANI     $01
2083  4625  32  7E  47   13.0          STA     TEMPP
2084  4628  BE           7.0           CMP     M
2085  4629  CA  38  46   10.0          JZ      EU1CN1
2086  462C                             ROUT    ADATA
 (1)  462C  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  462E  7F           4.0                   MOV     A,A             ;RETRY LINK
2087  462F  7E           7.0           MOV     A,M
2088  4630                             ROUT    EDATA
 (1)  4630  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4632  7F           4.0                   MOV     A,A             ;RETRY LINK
2089  4633                             ERRB    UNIT0,EU1CN1
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4633  CD  12  28   18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0010                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4636  10                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4637  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4638  CD  15  28   18.0          EU1CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  463B  DA  71  43   10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2090                             ;>SECOND GCR PARITY STATE NOT AS EXPECTED
2091                             ;>WHEN READ FROM 'WCSP'' BIT OF PORT 1 STATUS BYTE
2092
2093  463E                       S
 (1)                             ; ***************************************************************************
2094                             ;TAPE UNIT PORT 2 TEST - DATA PHASE ONE
2095  463E                       S
 (1)                             ; ***************************************************************************
2096
2097  463E  21  7B  47   10.0    EUNIT2: LXI     H,EGCRD
2098  4641  3A  76  47   13.0            LDA     UNITMP          ;GET THE UNIT MAP BYTE
2099  4644  E6  04       7.0             ANI     $04             ;UNIT 2 PRESENT?
2100  4646  CA  8E  46   10.0            JZ      EUNIT3          ;NO-CONTINUE
2101  4649  DB  E0       10.0            IN      INTSTA
2102  464B  E6  80       7.0             ANI     BIT7
2103  464D  F6  02       7.0             ORI     @02
2104  464F  D3  E0       10.0            OUT     MBSEL
2105  4651  DB  40       10.0            IN      TSTS            ;YES-GET THE TU STAEUS
2106  4653  32  7D  47   13.0            STA     TEMPD
2107  4656  BE           7.0             CMP     M               ;COMPARE WITH EXPECTED DATA
2108  4657  CA  66  46   10.0            JZ      EU2CN0          ;CONTINUE IF EQUAL
2109  465A                             ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  465A  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  465C  7F           4.0                   MOV     A,A             ;RETRY LINK
2110  465D  7E           7.0           MOV     A,M             ;GET EXPECTED DATA
2111  465E                             ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  465E  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4660  7F           4.0                   MOV     A,A             ;RETRY LINK
```

```
                                                ERRB    UNIT0,EU2CN0
                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                                EU2CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SECOND GCR DATA STATE NOT AS EXPECTED
                                        ;>WHEN READ FROM TJ2 STATUS BYTE

 2117  466C  21  7A  47      10.0              LXI     H,EGCRP
 2118  466F  DB  48         10.0              IN      PSTAT
 2119  4671  0F             4.0               RRC
 2120  4672  0F             4.0               RRC
 2121  4673  E6  01         7.0               ANI     $01
 2122  4675  32  7E  47      13.0              STA     TEMPP
 2123  4678  BE             7.0               CMP     M
 2124  4679  CA  88  46      10.0              JZ      EU2CN1
 2125  467C                                   ROUT    ADATA
  (1)  467C  D3  94         10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  467E  7F             4.0                       MOV     A,A             ;RETRY LINK
 2126  467F  7E             7.0               MOV     A,M
 2127  4680                                   ROUT    EDATA
  (1)  4680  D3  95         10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4682  7F             4.0                       MOV     A,A             ;RETRY LINK
 2128  4683                                   ERRB    UNIT0,EU2CN1
                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  4683  CD  12  28      18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)  4686  12                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4687  00                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4688  CD  15  28      18.0                     .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  468B  DA  71  43      10.0              EU2CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SECOND GCR PARITY STATE NOT AS EXPECTED
                                        ;>WHEN READ FROM 'WCSP'' BIT OF PORT 2 STATUS BYTE

 2132  468E                                   S
  (1)                                   ; ***************************************************************************************
 2133                                   ;TAPE UNIT PORT 3 TEST - DATA PHASE ONE
 2134  468E                                   S
  (1)                                   ; ***************************************************************************************

 2136  468E  21  7B  47      10.0      EUNIT3: LXI     H,EGCRD
 2137  4691  3A  76  47      13.0              LDA     UNITMP          ;GET THE UNIT MAP BYTE
 2138  4694  E6  08         7.0               ANI     $08             ;UNIT 3 PRESENT?
 2139  4696  CA  DE  46      10.0              JZ      PHASE3          ;NO-CONTINUE
 2140  4699  DB  E0         10.0              IN      INTSTA
 2141  469B  E6  80         7.0               ANI     BIT7
 2142  469D  F6  03         7.0               ORI     @03
 2143  469F  D3  E0         10.0              OUT     MBSEL
 2144  46A1  DB  40         10.0              IN      TSTS            ;YES-GET THE TU STAEUS
 2145  46A3  32  7D  47      13.0              STA     TEMPD
 2146  46A6  BE             7.0               CMP     M               ;COMPARE WITH EXPECTED DATA
```

```
                                        JZ      EU3CN0          ;CONTINUE IF EQUAL
                                        ROUT    ADATA           ;STORE ACTUAL DATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MOV     A,M             ;GET EXPECTED DATA
                                        ROUT    EDATA           ;STORE EXPECTED DATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    UNIT0,EU3CN0
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE                   ;PRINT ROUTINE NUMBER
                                        EU3CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>SECOND GCR DATA STATE NOT AS EXPECTED
                                ;>WHEN READ FROM TU3 STATUS BYTE

                                        LXI     H,EGCRP
                                        IN      PSTAT
                                        RRC
                                        RRC
                                        ANI     $01
                                        STA     TEMPP
                                        CMP     M
                                        JZ      EU3CN1
                                        ROUT    ADATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MOV     A,M
                                        ROUT    EDATA
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        ERRB    UNIT0,EU3CN1
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE                   ;PRINT ROUTINE NUMBER
                                        EU3CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>SECOND GCR PARITY STATE NOT AS EXPECTED
                                ;>WHEN READ FROM 'WCSP'' BIT OF PORT 3 STATUS BYTE


                                PHASE3: CALL    COMP            ;COMPLEMENT THE EXPECTED DATA
                                        LDA     PASSCNT         ;GET THE PASS COUNTER
                                        INR     A               ;UPDATE
                                        STA     PASSCNT         ;RESTORE IT
                                        LDA     TEPMSK          ;GET THE TRACK ENABLE PARITY MASK
                                        MOV     B,A             ;BUFFER IT
                                        XRA     A               ;CLEAR A
                                        STA     TEPMSK          ;CLEAR THE TRACK ENABLE PARITY MASK
                                        MOV     A,B             ;GET THE MASK BACK
```

```
2179  46F1  A7                    4.0          ANA     A               ;SET THE CONDITION CODE
2180  46F2  CA  F6  46           10.0          JZ      PH31            ;CONTINUE WITH C BIT CLEAR
2181  46F5  37                    4.0          STC                     ;ELSE-SET THE C BIT
2182  46F6  3A  77  47           13.0  PH31:   LDA     TEDMSK          ;GET THE TRACK ENABLE DATA MASK
2183  46F9  1F                    4.0          RAR                     ;SHIFT 9 BIT MASK RIGHT
2184  46FA  32  77  47           13.0          STA     TEDMSK          ;SAVE THE NEW TRACK ENABLE DATA MASK
2185  46FD  A7                    4.0          ANA     A               ;SET CONDITION BITS
2186  46FE  C2  F3  43           10.0          JNZ     ENLP0           ;CONTINUE UNTIL MASK = 0
2187
2188  4701                                EXIT:   ENDTST  UNIT0
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4701                              REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4701  CD  06  28       18.0                        CALL    REQST
 (2)  4704  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4705  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  4707  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4709  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  470A  07                                           .BYTE   7               ;REQUEST CODE
 (1)  470B  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  470E  3D                    4.0      DCR     A               ;DOWNCOUNT
 (1)  470F  32  9A  4F       13.0          STA     ITERA           ;SAVE COUNT
 (1)  4712  F2  71  43       10.0          JP      UNIT0           ;DO TEST UNTIL TILL = 0
2189  4715  C3  18  28       10.0          JMP     TSTEND
2190
2191  4718                                S
 (1)                                    ; ***************************************************************************
2192                                    ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
2193                                    ;       'CLK' BIT TO A '1' THEN A '0'.
2194  4718                              S
 (1)                                    ; ***************************************************************************
2195
2196  4718  3A  7C  47           13.0  CLKSYS: LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
2197  471B  F6  40                7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
2198  471D  D3  F0               10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
2199  471F  E6  BF                7.0          ANI     @277            ;STRIP OFF CLOCK BIT
2200  4721  D3  F0               10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
2201  4723  C9                   10.0          RET                     ;EXIT-CLOCK CYCLE
2202
2203
2204  4724  3A  7B  47           13.0  COMP:   LDA     EGCRD           ;COMPLEMENT THE DATA
2205  4727  2F                    4.0          CMA
2206  4728  32  7B  47           13.0          STA     EGCRD
2207  472B  3A  7A  47           13.0          LDA     EGCRP           ;COMPLEMENT THE PARITY
2208  472E  2F                    4.0          CMA
2209  472F  E6  01                7.0          ANI     $01
2210  4731  32  7A  47           13.0          STA     EGCRP
2211  4734  C9                   10.0          RET
2212
2213  4735  21  78  47           10.0  ADJDAT: LXI     H,TEPMSK        ;GET THE ADDRESS OF THE PARITY TRACK ENABLE BIT
2214  4738  3A  7A  47           13.0          LDA     EGCRP           ;GET THE EXPECTED PARITY BIT
2215  4738  A6                    7.0          ANA     M               ;AND WITH THE TRACK ENABLE MASK
2216  473C  32  7A  47           13.0          STA     EGCRP           ;SAVE THE NEW EXPECTED PARITY BIT
2217  473F  21  77  47           10.0          LXI     H,TEDMSK        ;GET THE ADDRESS OF THE DATA TRACK ENABLE BITS
2218  4742  3A  7B  47           13.0          LDA     EGCRD           ;GET THE EXPECTER DATA BITS
```

```
2219  4745  A6                    7.0        ANA     M              ;AND WITH THE TRACK ENABLE MASK
2220  4746  32   7B   47          13.0        STA     EGCRD          ;SAVE THE NEW EXPECTED DATA BITS
2221  4749  C9                    10.0        RET
```

```
2223                                    .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
2224  474A                              SSUB
 (1)                                    ; ************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
2225                                    ;*CLEAR ALL TU PORTS
2226  474A                              SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2227                                    ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
2228                                    ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
2229                                    ;*AND LOOP MODES.
2230  474A                              SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2231                                    ;*BGNSUB
2232                                    ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
2233                                    ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
2234                                    ;*     CLEAR PORT SELECT FOR TRANSPORT
2235                                    ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
2236                                    ;*     CLEAR PORT DIAGNOSTIC CONTROL
2237                                    ;*     CLEAR PORT AMTIE WORD
2238                                    ;*ENDSUB
2239  474A                              S
 (1)                                    ; ************************************************************************
2240  474A  F5              12.0       CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
2241  474B  C5              12.0               PUSH    B
2242  474C  06  00           7.0               MVI     B,0             ;START TO CLEAR AT PORT #0
2243  474E  DB  E0           10.0       CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
2244  4750  E6  80           7.0               ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
2245  4752  B0              4.0               ORA     B               ;ADD IN THE SELECTED PORT #
2246  4753  D3  E0          10.0               OUT     MBSEL           ;RESET TO THIS PORT
2247  4755  3E  80           7.0               MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
2248  4757  D3  40          10.0               OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
2249  4759  AF              4.0               XRA     A               ;CLEAR TU COMMAND A
2250  475A  D3  40          10.0               OUT     TCMD            ;
2251  475C  3E  81           7.0               MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
2252  475E  D3  40          10.0               OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
2253  4760  3E  00           7.0               MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
2254  4762  D3  40          10.0               OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
2255  4764  AF              4.0               XRA     A
2256  4765  D3  44          10.0               OUT     TAMT            ;CLEAR AMTIE WORD
2257  4767  D3  48          10.0               OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
2258  4769  D3  4C          10.0               OUT     PENAB           ;CLEAR PORT ENABLE WORD
2259  476B  04              4.0               INR     B               ;POINT TO THE NEXT PORT TO CLEAR
2260  476C  78              4.0               MOV     A,B
2261  476D  FE  04           7.0               CPI     4               ;DONE?
2262  476F  C2  4E  47      10.0               JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
2263  4772  C1              10.0               POP     B               ;RESET B & C
2264  4773  F1              10.0               POP     PSW             ;ALL DONE
2265  4774  C9              10.0               RET                     ;EXIT
```

```
2267  4775   00                PASSCNT: .BYTE 0          ;NUMBER OF PASSES THROUGH THE TEST
2268  4776   00                UNITMP:  .BYTE   0        ;UNIT MAP
2269  4777   00                TEDMSK:  .BYTE   0        ;TRACK ENABLE DATA BIT MASK
2270  4778   00                TEPMSK:  .BYTE   0        ;TRACK ENABLE PARITY BIT MASK
2271  4779   00                TOUT:    .BYTE   0        ;TIMEOUT COUNT
2272  477A   00                EGCRP:   .BYTE   0        ;EXPECTED GCR PARITY STATE
2273  477B   00                EGCRD:   .BYTE   0        ;EXPECTED GCR DATA STATE
2274  477C   00                CCTLWD:  .BYTE   0        ;SOFTWARE CLOCK IMAGE
2275  477D   00                TEMPD:   .BYTE   0        ;TEMP. DATA STORAGE
2276  477E   00                TEMPP:   .BYTE   0        ;TEMP. PARITY STORAGE
2277         0000                       .END
```

```
A      =%0007          ADATA  = 0094          ADJDAT  4735          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090         BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004         BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040         BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089         CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B         CBYTL  = 008A         CCTLWD  477C
CDG1H  = 0087          CDG1L  = 0086         CDG2H  = 0093         CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094         CDVTH  = 008D         CDVTL  = 008C
CHPTIE= 0028           CH0TIE= 0020          CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027           CKLOP  = 2815         CLEAR   474A          CLKCTL= 00F0
CLKSYS  4718           CLOCK   4F26          CLRLP   474E          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B         CMC1L  = 009A         CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F         CMC3L  = 009E         CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7          COMP    4724          CRCWRD= 0018
CSAVE   4F9D           CSRLH  = 0091         CSRLL  = 0090         CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F         CTSTL  = 008E         CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083         CXINL  = 0082         C.     = 0001
C.AVAI= 0080           C.DP   = 0008         C.DSE  = 0010         C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070         C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080         C.SHR  = 0040         C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002         D      =%0002         DATACT= 00D0
DBUS    4F28           DBUSCT= 00C0          DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9         DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088         DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300           DIAGRM= 4F90          DIARD  = 000B         DONE1  = 0045
DONINT= 0010           DSAVE   4F9E          DSE    = 0006         DUMMY   4368  G
D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010          D.LAGC= 0020
D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042          ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095         EGCRD   477B
EGCRP   477A           ENCN0   442C  G       ENLP0   43F3          EOTCLR= 0003
ERFLG   4F93           ERLP   = 2809         ERLPA  = 280F         ERLPB  = 2812
ERLPE  = 280C          ERNUM   4F90          ERRCNT= 00D6          ESAVE   4F9F
EUNIT0  459A           EUNIT1  45EE          EUNIT2  463E          EUNIT3  468E
EU0CN0  45C6  G        EU0CN1  45E8  G       EU1CN0  4616  G       EU1CN1  4638  G
EU2CN0  4666  G        EU2CN1  4688  G       EU3CN0  46B6  G       EU3CN1  46D8  G
EXIT    4701           E.ACRC= 0010          E.AMT  = 0020         E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008          E.RPE  = 0040         E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004         FIFORD= 006A          FORMAT  4F25
FWDTST= 0061           GCRID  = 0089         GCRSET= 0002          GO      43C2
GOODTM= 0092           H      =%0004         HLSAVE  4FA0          IE     = 0008
INTSTA= 00E0           ITERA   4F9A          I.PWR  = 0020         I.RMPE= 0040
I5.5   = 0010          I6.5   = 0020         I7.5   = 0040         KCALL  = 005F
KCLR   = 007B          KDEP   = 003F         KENAB  = 0078         KEXAM  = 003E
KEYBRD= 00C8           KEY1   = 0078         KEY10  = 006D         KEY11  = 006E
KEY12  = 006F          KEY13  = 005C         KEY14  = 005D         KEY15  = 005E
KEY16  = 005F          KEY17  = 003C         KEY18  = 003D         KEY19  = 003E
KEY2   = 0079          KEY20  = 003F         KEY3   = 007A         KEY4   = 007B
```

```
KEY5   = 0074          KEY6   = 0075          KEY7   = 0076          KEY8   = 0077
KEY9   = 006C          KINTA  = 006F          KLDAD  = 003D          KN0    = 003C
KN1    = 005C          KN2    = 005D          KN3    = 005E          KN4    = 006C
KN5    = 006D          KN6    = 006E          KN7    = 0074          KN8    = 0075
KN9    = 0076          KU2    = 0079          KU3    = 007A          KU8    = 0077
L      =%0005          LBLANK= 000F           LCE    = 000B          LCH    = 000C
LCL    = 000D          LCP    = 000E          LC0    = 0000          LC1    = 0001
LC2    = 0002          LC3    = 0003          LC4    = 0004          LC5    = 0005
LC6    = 0006          LC7    = 0007          LC8    = 0008          LC9    = 0009
LDLEDA= 00CA           LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046          LPFLG    4F94
LPNUM    4F92          M      =%0006          MBSEL  = 00E0          MB.A   = 0008
MB.B   = 0004          MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000
MSE    = 0008          MSGN   = 0014          MTACLR= 0000           MT.ARA= 0020
MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008
MT.LWR= 0004           MT.MOT= 0002           MT.NWT= 0080           MT.PEC= 0040
MT.PSB= 0004           MT.PS0= 0001           MT.PS1= 0002           MT.REV= 0020
MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020
M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008
M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020
M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080           M.RDEN= 0002
M.RDPE= 0008           M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040
M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080
M5.5   = 0001          M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088
OKAY   = 00FF          OPRRAM= 4300           OPSTRT= 0058           OPVER  = 0040
PADCNT= 00D5           PADCRC= 0080           PASSCN   4775          PDIAG  = 0048
PEID   = 008A          PENAB  = 004C          PESET  = 0001          PHASE2   4579
PHASE3   46DE          PH2CN    4594  G       PH31     46F6          PL     = 00B1
PRDD   = 004C          PRENF  = 009C          PS     = 00B2          PSTAT  = 0048
PSW    =%0009          P.AMTP= 0001           P.BCTC= 0040           P.CMDP= 0020
P.INTE= 0080           P.LCS  = 0040          P.LWR  = 0020          P.RDP  = 0002
P.RPEN= 0004           P.RPST= 0002           P.RPOE= 0020           P.RP1E= 0010
P.RP2E= 0020           P.RP3E= 0010           P.SING= 0080           P.STAT= 0002
P.STPE= 0080           P.TACH= 0008           P.TUPR= 0010           P.WCSP= 0004
P.WDS  = 0040          P.WFLP= 0001           P.WPEN= 0010           P.WPOE= 0008
P.WP1E= 0004           P.WP2E= 0008           P.WP3E= 0004           P.5VOK= 0002
QUE    = 281B          QUEM   = 281E          RAMT   = 0010          RARA   = 0006
RARAI  = 0004          RCHBD0= 0048           RCHBD1= 0047           RCHOK  = 0046
RCHTST= 000C           RCLRT  = 000D          RCMD   = 000B          RCMLP  = 0003
RCONT  = 0080          RDATA  = 0017          RDCLK  = 0010          RDON   = 0011
READG  = 0007          REND   = 0014          REQST  = 2806          RESCHR= 00D1
REVTST= 0064           REWIND= 0004           RFIFOL= 0008           RGCLK  = 0002
RGCRI  = 0003          RIBG   = 0001          RILL   = 0012          RINST  = 000C
RMCTST= 0008           RMK2   = 0013          RNOP   = 0000          RPATH  = 0001
RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003          RPCTL  = 0009
RPEI   = 0002          RPFAIL= 0000           RPF1   = 009D          RPF2   = 009E
RPOK   = 0043          RPOSTN= 0016           RPSTA  = 0015          RRCMT  = 000A
RSTAT  = 0002          RTIEB  = 000A          RTIER  = 0030          RTM    = 0005
RUNKI  = 0009          RUPTST= 005E           RWDUNL= 0005           R.AMT  = 0001
R.BOP  = 0008          R.DATA= 0040           R.DON  = 0002          R.DRDY= 0010
R.END  = 0010          R.ILL  = 0004          R.JVOK= 0004           R.MK2  = 0008
R.PLOD= 0008           R.PLO0= 0010           R.PLO1= 0020           R.POST= 0020
```

```
R.STNM= 0002          R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080
R.TSTD= 0040          R.VOK = 0080          R00H  = 0081          R00L  = 0080
R01H  = 0083          R01L  = 0082          R02H  = 0085          R02L  = 0084
R03H  = 0087          R03L  = 0086          R04H  = 0089          R04L  = 0088
R05H  = 0088          R05L  = 008A          R06H  = 008D          R06L  = 008C
R07H  = 008F          R07L  = 008E          R10H  = 0091          R10L  = 0090
R11H  = 0093          R11L  = 0092          R12H  = 0095          R12L  = 0094
R13H  = 0097          R13L  = 0096          R14H  = 0099          R14L  = 0098
R15H  = 009B          R15L  = 009A          R16H  = 009D          R16L  = 009C
R17H  = 009F          R17L  = 009E          R7.5  = 0010          SELCLR= 0000
SETATA= 00A1          SID   = 0080          SOD   = 0080          SOE   = 0040
SP    =%0008          SSCLK = 0040          SSTEP = 0005          STACK = 4FFF
STATRM= 4F20          STPCT   4F20          STRSP = 5000          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEDMSK  4777
TEMP    4F99          TEMPD   477D          TEMPP   477E          TEPMSK  4778
TEST1   4300          TEST2   434A          TMF   = 0099          TMRDY = 0040
TOUT    4779          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TUNIT0  4435          TUNIT1  4489          TUNIT2  44D9
TUNIT3  4529          TUSEL0= 00D1          TUSEL1= 00D2          TU0CN0  4461  G
TU0CN1  4483  G       TU1CN0  44B1  G       TU1CN1  44D3  G       TU2CN0  4501  G
TU2CN1  4523  G       TU3CN0  4551  G       TU3CN1  4573  G       TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002          UIBG  = 00A1
UNITMP  4776          UNIT0   4371          UNIT1   4386          UNIT2   439A
UNIT3   43AE          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN- 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          XLCN1   4330  G       XLLP1   4312          X.DONN= 0080
X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001
Y     =%000B          .     = 477F


ERRORS DETECTED:  0

*XMC1.A78/PTP,XMC1=NLIST,PARAM,MACRO,LIST,XMC1
RUN-TIME: 5 5 0 SECONDS
CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:09 SEQ 0434
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  XMC2 - TRANSLATOR MICRO CONTROLLER PART #2
                                .SBTTL  TEST 1 - XMC - PE - PREAMBLE/POSTAMBLE TEST
                                ;ID     XMC2-TRANSLATOR MICRO CONTROLLER PART #2
4300                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - PE - PREAMBLE/POSTAMBLE TEST
4300                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A PE RECORD OF ONE BYTE
                                ;AND THEN SINGLE STEPS THE WRITE PATH (XMC, XMC) WHILE MONITORING THE
                                ;XMC 'XL WCLK'' BIT.  THE OUTPUT OF THE TRANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, DATA, AND POSTAMBLE.  THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT,
                                ;DATA STATES AND FLUX REVERSAL RATE.
4300                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   INITIALIZE THE BYTE, PAD AND ERROR COUNTERS
                                ;*   CLEAR THE ERROR COUNTER
                                ;*   CALL SUBROUTINE CLEAR
                                ;*   GET USER SELECTED TU PORT FOR TEST
                                ;*   ISSUE RESTART TO THE XMC CONTROL BYTE
                                ;*   SET THE SINGLE STEP CLOCK
                                ;*   CLEAR RESTART TO THE XMC CONTROL BYTE
                                ;*   SET WRITE PATH ENABLE TO SELECTED TU PORT
                                ;*   LOAD 300(8) IN THE RESIDUAL CHARACTER BYTE
                                ;*   SET FORMAT TO PDP-11 NORMAL
                                ;*   LOAD 1 IN THE BYTE COUNTER (BITS 7:0)
                                ;*   LOAD 0 IN THE BYTE COUNTER (BITS 15:8)
                                ;*   CLEAR THE TRACK ENABLE BYTES
                                ;*   SET THE DDR CONTROL TO 'OUT''
                                ;*   LOAD DDRA WITH 0
                                ;*   LOAD DDRB WITH 374(8)
                                ;*   LOAD DDRC WITH 3
                                ;*   SET ALL TRACK ENABLE BITS
                                ;*   CLEAR THE POINTER COUNTER
                                ;*   INITIALIZE EXPECTED RESULTS POINTER
                                ;*   SET LOOP COUNT TO 377(8)
                                ;*   BGNDO
                                ;*   :  SET SUBROUTINE CLKSYS
                                ;*   :  DECREMENT THE LOOP COUNT
                                ;*   :  DO UNTIL THE LOOP COUNT=0
                                ;*   ENDDO
                                ;*   IF XMC ROM PARITY ERROR SET
                                ;*   :  THEN-ERROR-RESTART OF XMC NOT SUCCESSFUL
                                ;*   :  ELSE-CONTINUE
                                ;*   ENDIF
                                ;*   ISSUE XMC ENABLE, XMC ENABLE AND WRITE TO THE XMC CONTROL BYTE
                                ;*   SET THE LOOP COUNT TO 377(8)
```

```
1374                    ;*  BGNDO
1375                    ;*  :  BGNDO
1376                    ;*  :  :  DO WHILE LOOP COUNT NOT=0 AND WRITE DATA STROBE=0
1377                    ;*  :  :  DECREMENT THE LOOP COUNT
1378                    ;*  :  ENDDO
1379                    ;*  :  IF LOOP COUNT=0
1380                    ;*  :  :  THEN-ERROR-TIMEOUT WAITING FOR WDS TO SET
1381                    ;*  :  :  ELSE-CONTINUE
1382                    ;*  :  ENDIF
1383                    ;*  :  CALL SUBROUTINE CLKSYS
1384                    ;*  :  CALL SUBROUTINE CLKSYS
1385                    ;*  :  IF WRITE DATA STROBE=0
1386                    ;*  :  :  THEN-CONTINUE
1387                    ;*  :  :  ELSE-ERROR-WRITE DATA STROBE DID NOT CLEAR
1388                    ;*  :  ENDIF
1389                    ;*  :  CALL SUBROUTINE COMPPE
1390                    ;*  :  INPUT THE ACTUAL TRANSLATOR OUTPUT FROM USER SELECTED TU PORT
1391                    ;*  :  GET THE EXPECTED TRANSLATOR OUTPUT FROM THE TABLE
1392                    ;*  :  IF ACTUAL DATA=EXPECTED DATA
1393                    ;*  :  :  THEN-CONTINUE
1394                    ;*  :  :  ELSE-ERROR-PE PARITY BIT CELL INCORRECT
1395                    ;*  :  ENDIF
1396                    ;*  :  IF WRITE MICRO STATUS=PE PARITY ERROR OR XMC ROM PARITY ERROR
1397                    ;*  :  :  THEN-ERROR-PE PARITY ERROR OR ROM PARITY ERROR
1398                    ;*  :  :  ELSE-CONTINUE
1399                    ;*  :  ENDIF
1400                    ;*  :  INCREMENT THE EXPECTED DATA TABLE POINTER
1401                    ;*  :  INCREMENT THE POINTER COUNTER
1402                    ;*  :  SET THE LOOP COUNT=56(8)
1403                    ;*  :  BGNDO
1404                    ;*  :  :  CALL SUBROUTINE CLKSYS
1405                    ;*  :  :  DECREMENT THE LOOP COUNT
1406                    ;*  :  :  DO UNTIL LOOP COUNT=0 OR WRITE DATA STROBE=1
1407                    ;*  :  ENDDO
1408                    ;*  :  IF LOOP COUNT=0
1409                    ;*  :  :  THEN-IF POINTER COUNT NOT=80 OR 96
1410                    ;*  :  :        :  THEN-ERROR-IMPROPER FLUX RATE
1411                    ;*  :  :        :        CALL SUBROUTINE COMPPE
1412                    ;*  :  :        :  ELSE-CONTINUE
1413                    ;*  :  :       ENDIF
1414                    ;*  :  :  ELSE-CONTINUE
1415                    ;*  :  ENDIF
1416                    ;*  :  CALL SUBROUTINE CLKSYS
1417                    ;*  :  CALL SUBROUTINE CLKSYS
1418                    ;*  :  CALL SUBROUTINE CLKSYS
1419                    ;*  :  IF WRITE DATA STROBE=0
1420                    ;*  :  :  THEN-CONTINUE
1421                    ;*  :  :  ELSE-ERROR-WDS DID NOT RESET
1422                    ;*  :  ENDIF
1423                    ;*  :  INPUT THE ACTUAL TRANSLATOR OUTPUT AT SELECTED TU PORT
1424                    ;*  :  GET THE EXPECTED TRANSLATOR OUTPUT FROM THE TABLE
1425                    ;*  :  IF ACTUAL DATA=EXPECTED DATA
1426                    ;*  :  :  THEN-CONTINUE
1427                    ;*  :  :  ELSE-ERROR-PE BIT CELL INCORRECT
```

```
                                ;*  :  ENDIF
                                ;*  :  INPUT THE ACTUAL TRANSLATOR PARITY FROM SELECTED TU PORT
                                ;*  :  GET THE EXPECTED TRANSLATOR PARITY FROM THE TABLE
                                ;*  :  IF ACTUAL PARITY=EXPECTED PARITY
                                ;*  :  :  THEN-CONTINUE
                                ;*  :  :  ELSE-ERROR-PE PARITY BIT CELL INCORRECT
                                ;*  :  ENDIF
                                ;*  :  IF WRITE MICRO STATUS=PE PARITY ERROR OR XMC ROM PARITY ERROR
                                ;*  :  :  THEN-ERROR-PE PARITY ERROR OR ROM PARITY ERROR
                                ;*  :  :  ELSE-CONTINUE
                                ;*  :  ENDIF
                                ;*  :  IF THE POINTER COUNTER NOT=TABLE LENGTH
                                ;*  :  :  THEN-INCREMENT THE POINTER COUNTER
                                ;*  :  :       INCREMENT THE TABLE POINTER
                                ;*  :  :       SET THE LOOP COUNT TO 46
                                ;*  :  :  ELSE-CONTINUE
                                ;*  :  ENDIF
                                ;*  :  DO UNTIL THE POINTER COUNTER=TABLE LENGTH
                                ;*  ENDDO
                                ;*  IF TRANSLATOR NOT DONE=0
                                ;*  :  THEN-ERROR-TRANSLATOR FINISHED BEFORE WRITE MICRO
                                ;*  :  ELSE-IF WRITE MICRO NOT DONE=0
                                ;*  :       :  THEN-CONTINUE
                                ;*  :       :  ELSE-ERROR-WRITE MICRO NOT DONE
                                ;*  :       ENDIF
                                ;*  :       SET THE LOOP COUNT TO 20
                                ;*  :       BGNDO
                                ;*  :       :  CALL SUBROUTINE CLKSYS
                                ;*  :       :  DECREMENT THE LOOP COUNT
                                ;*  :       :  DO UNTIL LOOP COUNT=0 OR XMC DONE
                                ;*  :       ENDDO
                                ;*  :       IF LOOP COUNT=0
                                ;*  :       :  THEN-ERROR-XMC DID NOT FINISH
                                ;*  :       :  ELSE-CONTINUE
                                ;*  :       ENDIF
                                ;*  ENDIF
                                ;*ENDTST
4300                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*XMC2 MICRO TEST 01
                                ;*XMC2 MICRO ERROR 01
                                ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
                                ;*M8958, M8955'S, M8959
                                ;*OPERATOR ERROR NO TM78 UNITS SELECTED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*XMC2 MICRO TEST 01
                                ;*XMC2 MICRO ERROR 02
                                ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
                                ;*M8958, M89[illegible]'S, M8959
                                ;*RESTART OF  MC DID NOT CLEAR XMC ROM ERROR BIT
                                ;*ACTUAL   NNNN
```

```
1479                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1480                                    ;*
1481                                    ;*XMC2 MICRO TEST 01
1482                                    ;*XMC2 MICRO ERROR 03
1483                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1484                                    ;*M8958, M8955'S, M8959
1485                                    ;*TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 2ND HALF PE BIT CELL TIME
1486                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1487                                    ;*FATAL ERROR - TEST ABORTED
1488                                    ;*
1489                                    ;*XMC2 MICRO TEST 01
1490                                    ;*XMC2 MICRO ERROR 04
1491                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1492                                    ;*M8958, M8955'S, M8959
1493                                    ;*2ND HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1494                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1495                                    ;*
1496                                    ;*XMC2 MICRO TEST 01
1497                                    ;*XMC2 MICRO ERROR 05
1498                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1499                                    ;*M8958, M8955'S, M8959
1500                                    ;*2ND HALF OF PE BIT CELL INCORRECT
1501                                    ;*ACTUAL = NNNN
1502                                    ;*EXPECTED = NNNN
1503                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1504                                    ;*
1505                                    ;*XMC2 MICRO TEST 01
1506                                    ;*XMC2 MICRO ERROR 06
1507                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1508                                    ;*M8958, M8955'S, M8959
1509                                    ;*2ND HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1510                                    ;*ACTUAL = NNNN
1511                                    ;*EXPECTED = NNNN
1512                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1513                                    ;*
1514                                    ;*XMC2 MICRO TEST 01
1515                                    ;*XMC2 MICRO ERROR 07
1516                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1517                                    ;*M8958, M8955'S, M8959
1518                                    ;*2ND HALF PE BIT CELL TIME - DETECTED PE PARITY E' ?OR OR
1519                                    ;*XMC ROM PARITY ERROR
1520                                    ;*ACTUAL = NNNN
1521                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1522                                    ;*
1523                                    ;*XMC2 MICRO TEST 01
1524                                    ;*XMC2 MICRO ERROR 10
1525                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1526                                    ;*M8958, M8955'S, M8959
1527                                    ;*TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 1ST HALF PE BIT CELL TIME
1528                                    ;*IMPROPER FLUX REVERSAL RATE WILL OCCUR ON TAPE
1529                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1530                                    ;*
1531                                    ;*XMC2 MICRO TEST 01
1532                                    ;*XMC2 MICRO ERROR 11
```

```
1533                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1534                    ;*M8958, M8955'S, M8959
1535                    ;*1ST HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1536                    ;*BYTE/SCLK COUNT NUMBER = LLL
1537                    ;*
1538                    ;*XMC2 MICRO TEST 01
1539                    ;*XMC2 MICRO ERROR 12
1540                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1541                    ;*M8958, M8955'S, M8959
1542                    ;*1ST HALF OF PE BIT CELL INCORRECT
1543                    ;*ACTUAL = NNNN
1544                    ;*EXPECTED = NNNN
1545                    ;*BYTE/SCLK COUNT NUMBER = LLL
1546                    ;*
1547                    ;*XMC2 MICRO TEST 01
1548                    ;*XMC2 MICRO ERROR 13
1549                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1550                    ;*M8958, M8955'S, M8959
1551                    ;*1ST HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1552                    ;*ACTUAL = NNNN
1553                    ;*EXPECTED = NNNN
1554                    ;*BYTE/SCLK COUNT NUMBER = LLL
1555                    ;*
1556                    ;*XMC2 MICRO TEST 01
1557                    ;*XMC2 MICRO ERROR 14
1558                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1559                    ;*M8958, M8955'S, M8959
1560                    ;*1ST HALF PE BIT CELL TIME - DETECTED EITHER PE PARITY ERROR OR
1561                    ;*XMC ROM PARITY ERROR
1562                    ;*ACTUAL = NNNN
1563                    ;*BYTE/SCLK COUNT NUMBER = LLL
1564                    ;*
1565                    ;*XMC2 MICRO TEST 01
1566                    ;*XMC2 MICRO ERROR 15
1567                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1568                    ;*M8958, M8955'S, M8959
1569                    ;*'XMC NOT DONE' =1 AT END OF XMC PE POSTAMBLE
1570                    ;*XMC SHOULD HAVE FINISHED BEFORE START OF PE POSTAMBLE
1571                    ;*BYTE/SCLK COUNT NUMBER = LLL
1572                    ;*
1573                    ;*XMC2 MICRO TEST 01
1574                    ;*XMC2 MICRO ERROR 16
1575                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1576                    ;*M8958, M8955'S, M8959
1577                    ;*XMC FAILED TO FINISH WITHIN SEVERAL CLOCKS PAST END OF PE POSTAMBLE
1578                    ;*'XMC NOT DONE' =1
1579                    ;*BYTE/SCLK COUNT NUMBER = LLL
1580                    ;*
1581                    ;*XMC2 MICRO TEST 01
1582                    ;*XMC2 MICRO ERROR 17
1583                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1584                    ;*M8958, M8955'S, M8959
1585                    ;*XMC FINISHED ('XMC NOT DONE' = 0) BEFORE THE XMC
1586                    ;*BYTE/SCLK COUNT NUMBER = LLL
```

```
1587  4300                                          S
 (1)                                                ; ***************************************************************************
1588
1589  4300                                          TEST1:  TESTX   @1
 (1)  4300  3E  01                        7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28                   18.0                 CALL    TSET            ;SETUP THE TEST
1590                                                ;%XMC2-PE-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1591                                                ;&M8958, M8955'S, M8959
1592  4305  AF                         4.0         XRA     A               ;CLEAR ALL TRACK ENABLE LINES
1593  4306  D3  D2                   10.0         OUT     TRKENA
1594  4308  D3  D2                   10.0         OUT     TRKENA
1595  430A                                        INICNT
 (1)  430A  3E  30                        7.0                 MVI     A,@060          ;INIT THE
 (1)  430C  D3  D7                   10.0                 OUT     CNTCTL          ;BYTE COUNTER
 (1)  430E  3E  70                        7.0             MVI     A,@160          ;INIT THE
 (1)  4310  D3  D7                   10.0                 OUT     CNTCTL          ;PAD COUNTER
 (1)  4312  3E  B0                        7.0             MVI     A,@260          ;INIT THE
 (1)  4314  D3  D7                   10.0                 OUT     CNTCTL          ;ERROR COUNTER
1596  4316                                        CLRECT
 (1)  4316  AF                         4.0                 XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4317  D3  D6                   10.0                 OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4319  D3  D6                   10.0                 OUT     ERRCNT          ;CLEAR BITS 15-8
1597  431B  CD  8B  45                 18.0         CALL    CLEAR
1598  431E                                        REQ     @7,0,0,0
 (1)  431E  CD  06  28                   18.0                 CALL    REQST
 (1)  4321  00                                               .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4322  00  00                                           .WORD   0               ;SYSTEM '0' COUNT
 (1)  4324  00  00                                           .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4326  00                                               .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  4327  07                                               .BYTE   @7              ;REQUEST CODE
1599  4328                                        RIN     R12L
 (1)  4328  DB  94                       10.0                IN      R12L                    ;READ R12L INTO AC
 (1)  432A  7F                            4.0                MOV     A,A                     ;RETRY LINK
1600  432B  32  66  46                   13.0         STA     UNITMP
1601  432E  06  00                        7.0         MVI     B,0
1602  4330  E6  01                        7.0         ANI     $01                     ;UNIT 0 PRESENT?
1603  4332  C2  5E  43                   10.0         JNZ     FOUND                   ;YES-USE IT
1604  4335  04                            4.0         INR     B
1605  4336  3A  66  46                   13.0         LDA     UNITMP                  ;
1606  4339  E6  02                        7.0         ANI     $02                     ;UNIT 1 PRESENT?
1607  433B  C2  5E  43                   10.0         JNZ     FOUND                   ;YES-USE IT
1608  433E  04                            4.0         INR     B
1609  433F  3A  66  46                   13.0         LDA     UNITMP                  ;
1610  4342  E6  04                        7.0         ANI     $04                     ;UNIT 2 PRESENT?
1611  4344  C2  5E  43                   10.0         JNZ     FOUND                   ;YES-USE IT
1612  4347  04                            4.0         INR     B
1613  4348  3A  66  46                   13.0         LDA     UNITMP
1614  434B  E6  08                        7.0         ANI     $08                     ;UNIT 3 PRESENT
1615  434D  C2  5E  43                   10.0         JNZ     FOUND                   ;YES-USE IT
1616                                                                                  ;ELSE - OPERATOR ERROR
```

```
1618  4350                                        ERR     EXIT,DUMMY
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4350  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4353  01                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4354  00                                            .BYTE
 (1)  4355  CD  15  28      18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4358  DA  6B  45      10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1619                                    ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1620                                    ;>FATAL ERROR - TEST ABORTED
1621  435B  C3  6B  45      10.0                JMP     EXIT
1622
1623  435E  DB  E0           10.0  FOUND:  IN      INTSTA          ;GET THE INTERRUPT STATUS
1624  4360  E6  80            7.0          ANI     BIT7            ;SAVE MB PORT SELECT BIT
1625  4362  B0                4.0          ORA     B               ;OR IN TU PORT BITS
1626  4363  D3  E0           10.0          OUT     MBSEL           ;SELECT THE PORT
1627  4365  3E  40            7.0          MVI     A,P.LCS         ;KILL THE MTA BOARD
1628  4367  D3  48           10.0          OUT     PDIAG
1629  4369  AF                4.0          XRA     A
1630  436A  D3  40           10.0          OUT     TCMD
1631
1632  436C  3E  01            7.0  T1LOOP: MVI     A,W.RST         ;RESTART THE XMC
1633  436E  D3  D3           10.0          OUT     WMCCTL
1634  4370  3E  05            7.0          MVI     A,SSTEP         ;SET CLOCK TO SINGLE STEP
1635  4372  D3  F0           10.0          OUT     CLKCTL
1636  4374  AF                4.0          XRA     A
1637  4375  D3  D3           10.0          OUT     WMCCTL          ;FINISH THE RESTART
1638
1639  4377  3E  10            7.0          MVI     A,P.WPEN        ;SET WRITE PATH ENABLE
1640  4379  D3  4C           10.0          OUT     PENAB
1641  437B  3E  C0            7.0          MVI     A,@300          ;SET FOR A 6 BYTE XFR (PE)
1642  437D  D3  D1           10.0          OUT     RESCHR          ;LOAD THE RESIDUAL CHAR.
1643  437F  3E  00            7.0          MVI     A,$0            ;SET FOR 11-NORMAL MODE
1644  4381  D3  D0           10.0          OUT     DATACTL         ;LOAD DATA CONTROL WORD
1645  4383  3E  01            7.0          MVI     A,1             ;LOAD THE BYTE COUNTER
1646  4385  D3  D4           10.0          OUT     BYTCNT
1647  4387  AF                4.0          XRA     A
1648  4388  D3  D4           10.0          OUT     BYTCNT
1649  438A  D3  D2           10.0          OUT     TRKENA          ;INIT THE TRACKS
1650  438C  D3  D2           10.0          OUT     TRKENA
1651  438E  3E  88            7.0          MVI     A,@210          ;SET THE DDR CONTROL TO 'OUT'
1652  4390  D3  DB           10.0          OUT     DDRCTL
1653  4392  AF                4.0          XRA     A
1654  4393  D3  D8           10.0          OUT     DDRA            ;SET BITS 5-0,17-16 TO 0
1655  4395  3E  FC            7.0          MVI     A,@374
1656  4397  D3  D9           10.0          OUT     DDRB            ;SET BITS 13-8 TO 1'S & 7-6 TO 0
1657  4399  3E  03            7.0          MVI     A,3
1658  439B  D3  DA           10.0          OUT     DDRC            ;SET BITS 15-14 TO 1'S
1659                                  ;BYTE A IN 11-NORM = 000
1660                                  ;BYTE B IN 11-NORM = 377
1661  439D  3E  FF            7.0          MVI     A,@377
1662  439F  D3  D2           10.0          OUT     TRKENA          ;ENABLE ALL TRACKS
1663  43A1  D3  D2           10.0          OUT     TRKENA
1664
```

```
 1665  43A3  21  B6  45   10.0          LXI   H,PXPTBL          ;LOAD ADDRESS OF EXPECTED
 1666  43A6  22  67  46   16.0          SHLD  POINTER           ;RESULTS TABLE INTO POINTER
 1667  43A9  3E  01        7.0          MVI   A,1
 1668  43AB  32  69  46   13.0          STA   PNTCNT            ;SAVE THE COUNT
 1669  43AE                             ROUT  R05L              ;START AT PE CHARACTER #1
  (1)  43AE  D3  8A        10.0                OUT   R05L                ;WRITE AC INTO R05L
  (1)  43B0  7F             4.0                MOV   A,A                 ;RETRY LINK
 1670  43B1  AF             4.0          XRA   A
 1671  43B2                             ROUT  R05H
  (1)  43B2  D3  8B        10.0                OUT   R05H                ;WRITE AC INTO R05H
  (1)  43B4  7F             4.0                MOV   A,A                 ;RETRY LINK
 1672  43B5  32  6A  46   13.0          STA   PARBEXP           ;INIT THE EXPECTED PARITY BIT
 1673  43B8  06  FF        7.0          MVI   B,$FF
 1674
 1675  43BA  CD  6E  45   18.0  RESTCK: CALL  CLKSYS
 1676  43BD  05             4.0          DCR   B
 1677  43BE  C2  BA  43   10.0          JNZ   RESTCK
 1678  43C1  DB  E0        10.0          IN    INTSTA            ;CHECK TO SEE IF XMC PARITY ERROR SET
 1679  43C3  E6  01        7.0          ANI   X.ROME            ;XMC ROM PARITY ERROR?
 1680  43C5  CA  D0  43   10.0          JZ    RESTOK            ;JUMP IF RESTART WAS OK
 1681  43C8                             ROUT  ADATA
  (1)  43C8  D3  94        10.0                OUT   ADATA               ;WRITE AC INTO ADATA
  (1)  43CA  7F             4.0                MOV   A,A                 ;RETRY LINK
 1682  43CB                          ERRA  T1LOOP,RESTOK,1
  (1)                             ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' VALID
  (1)  43CB  CD  0F  28   18.0                CALL  ERLPA               ;PROCESS ERROR - DO 2.3
  (1)       0002                              MSGN  =     MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  43CE  02                               .BYTE MSGN                ;MESSAGE NUMBER ID
  (1)  43CF  01                               .BYTE 1                   ;PRINT ROUTINE NUMBER
  (1)  43D0  CD  15  28   18.0          RESTOK::      CALL  CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
  (1)  43D3  DA  6C  43   10.0                JC    T1LOOP              ;LOOP ADDRESS IF LOOP SPECIFIED
 1683                          ;>RESTART OF XMC DID NOT CLEAR XMC ROM ERROR BIT
 1684  43D6  3E  C8        7.0          MVI   A,W.ENAB!X.ENAB!W.WRITE
 1685  43D8  D3  D3        10.0         OUT   WMCCTL            ;START THE OPERATION
 1686
 1687  43DA  06  FF        7.0  CLK1:  MVI   B,$FF             ;SET TIMEOUT COUNT TO 256
 1688                                                          ;
 1689  43DC  CD  6E  45   18.0  CLK:   CALL  CLKSYS            ;SINGLE STEP THE WRITE PATH
 1690  43DF  DB  48        10.0         IN    PSTAT             ;GET THE WRITE DATA STROBE STATUS
 1691  43E1  E6  40        7.0          ANI   P.WDS             ;P.WDS SET?
 1692  43E3  C2  F8  43   10.0          JNZ   BGNPRE            ;YES-BEGINNING OF PREAMBLE
 1693  43E6  05             4.0          DCR   B                 ;NO-DECREMENT CLOCK
 1694  43E7  C2  DC  43   10.0          JNZ   CLK               ;CONTINUE UNTIL ZERO
 1695  43EA                          ERR   T1LOOP,ABOCN0,1
  (1)                             ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  43EA  CD  09  28   18.0                CALL  ERLP                ;PROCESS ERROR - DO 2.3
  (1)       0003                              MSGN  =     MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  43ED  03                               .BYTE MSGN                ;MESSAGE NUMBER ID
  (1)  43EE  01                               .BYTE 1
  (1)  43EF  CD  15  28   18.0          ABOCN0::      CALL  CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
  (1)  43F2  DA  6C  43   10.0                JC    T1LOOP              ;LOOP ADDRESS IF LOOP SPECIFIED
 1696                          ;>TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 2ND HALF PE BIT CELL TIME
 1697                          ;>FATAL ERROR - TEST ABORTED
 1698  43F5  C3  18  28   10.0          JMP   TSTEND
```

```
1699
1700  43F8  CD  6E  45   18.0  BGNPRE: CALL    CLKSYS          ;CLOCK THE BIT CLEAR
1701  43FB  CD  6E  45   18.0          CALL    CLKSYS
1702  43FE  DB  48       10.0          IN      PSTAT           ;GET WRT DATA STROBE STATUS WORD
1703  4400  E6  40        7.0          ANI     P.WDS           ;SEE IF BIT CLEARED AFTER CLOCK CYCLE
1704  4402  CA  0A  44   10.0          JZ      PREC1           ;JUMP IF OK
1705  4405                             ERR     T1LOOP,PREC1,1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4405  CD  09  28   18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4408  04                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4409  01                                .BYTE   1
 (1)  440A  CD  15  28   18.0          PREC1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  440D  DA  6C  43   10.0                  JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1706                          ;>2ND HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1707  4410  CD  79  45   18.0          CALL    COMPPE          ;COMPLEMENT THE EXPECTED PARITY BIT
1708  4413  DB  40       10.0          IN      TSTS            ;GET THE DATA READ
1709
1710  4415  47            4.0          MOV     B,A             ;SAVE IT IN B=ADATA
1711  4416                             ROUT    ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  4416  D3  94       10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4418  7F            4.0                  MOV     A,A             ;RETRY LINK
1712                                             ;
1713  4419  2A  67  46   16.0          LHLD    POINTER         ;GET POINTER TO EXPECTED DATA
1714  441C  7E            7.0          MOV     A,M             ;GET EXPECTED DATA BYTE
1715  441D  B8            4.0          CMP     B               ;COMPARE WITH ACTUAL
1716  441E  CA  29  44   10.0          JZ      PREC1A          ;CONTINUE IF EQUAL
1717                                             ;ELSE-ERROR
1718  4421                             ROUT    EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  4421  D3  95       10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4423  7F            4.0                  MOV     A,A             ;RETRY LINK
1719  4424                             ERRB    T1LOOP,PREC1A,1
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4424  CD  12  28   18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4427  05                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4428  01                                .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4429  CD  15  28   18.0          PREC1A::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  442C  DA  6C  43   10.0                  JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1720                          ;>2ND HALF OF PE BIT CELL INCORRECT
1721  442F  DB  48       10.0          IN      PSTAT           ;GET THE PORT PARITY BIT
1722  4431  E6  04        7.0          ANI     P.WCSP          ;SAVE ONLY THE PARITY BIT
1723  4433                             ROUT    ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  4433  D3  94       10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4435  7F            4.0                  MOV     A,A             ;RETRY LINK
1724  4436  21  6A  46   10.0          LXI     H,PARBEXP       ;POINT TO EXPECTED PARITY BIT
1725  4439  BE            7.0          CMP     M               ;ACTUAL = EXPECTED PARITY BIT?
1726  443A  CA  48  44   10.0          JZ      PREC2           ;JUMP IF OK
1727  443D  3A  6A  46   13.0          LDA     PARBEXP         ;GET THE EXPECTED PARITY BIT
1728  4440                             ROUT    EDATA
 (1)  4440  D3  95       10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4442  7F            4.0                  MOV     A,A             ;RETRY LINK
```

```
1730  4443                                                  ERRB    T1LOOP,PREC2,1
 (1)                                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4443  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4446  06                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4447  01                                             .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4448  CD  15  28        18.0                  PREC2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  444B  DA  6C  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1731                                            ;>2ND HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1732  444E  DB  E0              10.0                  IN      INTSTA          ;SEE IF ERROR DETECTED
1733  4450  E6  03               7.0                  ANI     X.PEPE!X.ROME
1734  4452  CA  5D  44          10.0                  JZ      PREC2A          ;JUMP IF OK
1735  4455                                            ROUT    ADATA
 (1)  4455  D3  94              10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4457  7F                   4.0                          MOV     A,A             ;RETRY LINK
1736  4458                                                  ERRA    T1LOOP,PREC2A,1
 (1)                                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4458  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0007                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  445B  07                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  445C  01                                             .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  445D  CD  15  28        18.0                  PREC2A::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4460  DA  6C  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1737                                            ;>2ND HALF PE BIT CELL TIME - DETECTED PE PARITY ERROR OR
1738                                            ;>XMC ROM PARITY ERROR
1739  4463  2A  67  46          16.0                  LHLD    POINTER         ;NO-UPDATE THE
1740  4466  23                   6.0                  INX     H               ;TABLE POINTER
1741  4467  22  67  46          16.0                  SHLD    POINTER         ;ADDRESS
1742  446A  3A  69  46          13.0                  LDA     PNTCNT          ;UPDATE THE POINTER COUNT
1743  446D  3C                   4.0                  INR     A
1744  446E  32  69  46          13.0                  STA     PNTCNT          ;SAVE NEW COUNT
1745  4471                                            ROUT    R05L            ;AND AGAIN
 (1)  4471  D3  8A              10.0                          OUT     R05L            ;WRITE AC INTO R05L
 (1)  4473  7F                   4.0                          MOV     A,A             ;RETRY LINK
1746
1747  4474  06  2E               7.0                  MVI     B,46            ;LOAD THE CLOCK COUNT
1748  4476  CD  6E  45          18.0          1$:     CALL    CLKSYS          ;CLOCK THE WRITE PATH
1749  4479  DB  48              10.0                  IN      PSTAT
1750  447B  E6  40               7.0                  ANI     P.WDS           ;SEE IF WCLK SET
1751  447D  C2  96  44          10.0                  JNZ     PREC3           ;JUMP IF ITS SET IN TIME
1752  4480  05                   4.0                  DCR     B               ;DECREMENT THE COUNT
1753  4481  C2  76  44          10.0                  JNZ     1$              ;CONTINUE UNTIL 23 CLOCKS
1754  4484  3A  69  46          13.0                  LDA     PNTCNT          ;SEE IF SPECIAL CASE...END OF PREAMBLE
1755  4487  FE  50               7.0                  CPI     80              ;40 PREAMBLE TICKS?
1756  4489  CA  9F  44          10.0                  JZ      PREC3A          ;JUMP IF END OF PREAMBLE
1757  448C  FE  60               7.0                  CPI     96              ;48 TICKS - AFTER ALL ONES ON POSTAMBLE
1758  448E  CA  9F  44          10.0                  JZ      PREC3A          ;JUMP IF END OF 1'S TICK IN POSTAMBLE
```

```
1760  4491                                          ERR     T1LOOP,PREC3,1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4491  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4494  08                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4495  01                                              .BYTE   1
 (1)  4496  CD  15  28        18.0                  PREC3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4499  DA  6C  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1761                                    ;>TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 1ST HALF PE BIT CELL TIME
1762                                    ;>IMPROPER FLUX REVERSAL RATE WILL OCCUR ON TAPE
1763  449C  CD  79  45        18.0              CALL    COMPPE          ;COMPLIMENT THE EXPECTED PARITY BIT
1764  449F  CD  6E  45        18.0      PREC3A: CALL    CLKSYS          ;NOW CLOCK IT CLEAR
1765  44A2  CD  6E  45        18.0              CALL    CLKSYS
1766  44A5  CD  6E  45        18.0              CALL    CLKSYS          ;THIS IS A HACK CLOCK CAUSE OF XMC LOGIC
1767  44A8  DB  48            10.0              IN      PSTAT           ;GET STATUS WORD
1768  44AA  E6  40             7.0              ANI     P.WDS           ;SEE IF IT CLEARED
1769  44AC  CA  B4  44        10.0              JZ      PREC3B          ;JUMP IF OK
1770  44AF                                    ERR     T1LOOP,PREC3B,1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44AF  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0009                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B2  09                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B3  01                                              .BYTE   1
 (1)  44B4  CD  15  28        18.0                  PREC3B::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44B7  DA  6C  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1771                                    ;>1ST HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1772  44BA  DB  40            10.0              IN      TSTS            ;GET THE DATA READ
1773  44BC  47                 4.0              MOV     B,A             ;SAVE IT IN B=ADATA
1774  44BD                                      ROUT    ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  44BD  D3  94            10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44BF  7F                 4.0                      MOV     A,A             ;RETRY LINK
1775                                                    ;
1776  44C0  2A  67  46        16.0              LHLD    POINTER         ;GET POINTER TO EXPECTED DATA
1777  44C3  7E                 7.0              MOV     A,M             ;GET EXPECTED DATA BYTE
1778  44C4  B8                 4.0              CMP     B               ;COMPARE WITH ACTUAL
1779  44C5  CA  D0  44        10.0              JZ      PRECN4          ;CONTINUE IF EQUAL
1780                                                    ;ELSE-ERROR
1781  44C8                                      ROUT    EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  44C8  D3  95            10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44CA  7F                 4.0                      MOV     A,A             ;RETRY LINK
1782  44CB                                      ERRB    T1LOOP,PRECN4,1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44CB  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000A                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CE  0A                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44CF  01                                              .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  44D0  CD  15  28        18.0                  PRECN4::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D3  DA  6C  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1783                                    ;>1ST HALF OF PE BIT CELL INCORRECT
1784  44D6  DB  48            10.0              IN      PSTAT           ;GET THE PORT PARITY BIT
1785  44D8  E6  04             7.0              ANI     P.WCSP          ;SAVE ONLY THE PARITY BIT
1786  44DA                                      ROUT    ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  44DA  D3  94            10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44DC  7F                 4.0                      MOV     A,A             ;RETRY LINK
```

```
1787  44DD  21  6A  46  10.0          LXI   H,PARBEXP        ;SET POINTER
1788  44E0  BE              7.0       CMP   M                ;ACTUAL = EXPECTED PARITY BIT?
1789  44E1  CA  EF  44      10.0      JZ    PREC5
1790  44E4  3A  6A  46      13.0      LDA   PARBEXP          ;GET THE EXPECTED PARITY BIT
1791  44E7                            ROUT  EDATA            ;SAVE FOR ERROR PRINTOUT
 (1)  44E7  D3  95            10.0             OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  44E9  7F                4.0              MOV   A,A            ;RETRY LINK
1792  44EA                      ERRB  T1LOOP,PREC5,1
 (1)                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44EA  CD  12  28  18.0             CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)  0008                               MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44ED  08                           .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  44EE  01                           .BYTE 1              ;PRINT ROUTINE NUMBER
 (1)  44EF  CD  15  28  18.0      PREC5:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44F2  DA  6C  43  10.0              JC    T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1793                          ;>1ST HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1794  44F5  DB  E0            10.0      IN    INTSTA           ;SEE IF ERROR DETECTED
1795  44F7  E6  03            7.0       ANI   X.PEPE!X.ROME
1796  44F9  CA  04  45        10.0      JZ    PREC5A           ;JUMP IF OK
1797  44FC                            ROUT  ADATA
 (1)  44FC  D3  94            10.0             OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  44FE  7F                4.0              MOV   A,A            ;RETRY LINK
1798  44FF                      ERRA  T1LOOP,PREC5A,1
 (1)                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  44FF  CD  0F  28  18.0             CALL  ERLPA          ;PROCESS ERROR - DO 2.3
 (1)  000C                               MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4502  0C                           .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  4503  01                           .BYTE 1              ;PRINT ROUTINE NUMBER
 (1)  4504  CD  15  28  18.0      PREC5A::        CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4507  DA  6C  43  10.0              JC    T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1799                          ;>1ST HALF PE BIT CELL TIME - DETECTED EITHER PE PARITY ERROR OR
1800                          ;>XMC ROM PARITY ERROR
1801  450A  3A  69  46  13.0          LDA   PNTCNT           ;GET THE CURRENT PAIR COUNT
1802  450D  06  B0            7.0       MVI   B,<PTBLEN-PXPTBL> ;CALC END OF TEST...
1803  450F  B8                4.0       CMP   B              ;HAVE ALL POSTAMBLE YET?
1804  4510  CA  29  45        10.0      JZ    RECEND         ;YES - SEE IF XMC & XMC ARE FINISHED
1805  4513  2A  67  46        16.0      LHLD  POINTER        ;GET THE TABLE POINTER
1806  4516  23                6.0       INX   H              ;UPDATE POINTER
1807  4517  22  67  46        16.0      SHLD  POINTER        ;SAVE IT
1808  451A  3A  69  46        13.0      LDA   PNTCNT         ;GET CURRENT COUNT
1809  451D  3C                4.0       INR   A              ;UPDATE IT
1810  451E  32  69  46        13.0      STA   PNTCNT         ;SAVE IT
1811  4521                            ROUT  R05L             ;SAVE AGAIN FOR ERROR PRINTING
 (1)  4521  D3  8A            10.0             OUT   R05L           ;WRITE AC INTO R05L
 (1)  4523  7F                4.0              MOV   A,A            ;RETRY LINK
1812
1813  4524  06  2E            7.0       MVI   B,46           ;RESET THE CLOCK COUNT FOR NEXT CYCLE
1814  4526  C3  DC  43        10.0      JMP   CLK            ;BACK TO LOOP
1815
1816  4529                    RECEND:
1817  4529  DB  D0            10.0      IN    WMCSTA         ;GET XMC STATUS
1818  452B  E6  80            7.0       ANI   X.DONN         ;XMC DONE
1819  452D  CA  60  45        10.0      JZ    SEQERR         ;YES-ERROR (OUT OF SEQUENCE)
1820  4530  DB  D0            10.0      IN    WMCSTA         ;INPUT THE XMC STATUS
```

```
1821  4532  E6  40          7.0          ANI     W.DONN          ;XMC DONE?
1822  4534  CA  3C  45      10.0          JZ      RECCN1          ;YES - SHOULD HAVE FINISHED AT POSTAMBLE
1823  4537                               ERR     T1LOOP,RECCN1,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4537  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000D                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453A  0D                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453B  01                                          .BYTE   1
 (1)  453C  CD  15  28      18.0                RECCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  453F  DA  6C  43      10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1824                                    ;>'XMC NOT DONE' =1 AT END OF XMC PE POSTAMBLE
1825                                    ;>XMC SHOULD HAVE FINISHED BEFORE START OF PE POSTAMBLE
1826  4542  06  14          7.0          MVI     B,20            ;LOAD THE TIMEOUT COUNT
1827  4544  CD  6E  45      18.0  3$:     CALL    CLKSYS          ;CLOCK THE SYSTEM
1828  4547  DB  D0          10.0          IN      WMCSTA          ;INPUT THE XMC STATUS
1829  4549  E6  80          7.0          ANI     X.DONN          ;XMC-DONE?
1830  454B  CA  57  45      10.0          JZ      RECCN3          ;YES-EXIT
1831  454E  05              4.0          DCR     B               ;NO-DECREMENT THE COUNT
1832  454F  C2  44  45      10.0          JNZ     3$              ;CONTINUE IF NOT ZERO
1833  4552                               ERR     T1LOOP,RECCN3,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4552  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000E                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4555  0E                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4556  01                                          .BYTE   1
 (1)  4557  CD  15  28      18.0                RECCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  455A  DA  6C  43      10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1834                                    ;>XMC FAILED TO FINISH WITHIN SEVERAL CLOCKS PAST END OF PE POSTAMBLE
1835                                    ;>'XMC NOT DONE' =1
1836  455D  C3  6B  45      10.0          JMP     EXIT
1837
1838  4560                         SEQERR: ERR     T1LOOP,RECCN2,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4560  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000F                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4563  0F                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4564  01                                          .BYTE   1
 (1)  4565  CD  15  28      18.0                RECCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4568  DA  6C  43      10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1839                                    ;>XMC FINISHED ('XMC NOT DONE' = 0) BEFORE THE XMC
1840  456B  C3  18  28      10.0  EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLKSYS
                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLOCK THE SYSTEM
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE ISSUES ONE SINGLE STEP CLOCK
456E  3E  05              7.0   CLKSYS: MVI     A,SSTEP          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
4570  F6  40              7.0           ORI     SSCLK            ;ADD IN 'CLK' BIT
4572  D3  F0             10.0           OUT     CLKCTL           ;LOAD CLOCK CONTROL
4574  E6  BF              7.0           ANI     @277             ;STRIP OFF CLOCK BIT
4576  D3  F0             10.0           OUT     CLKCTL           ;LOAD CLOCK CONTROL WORD
4578  C9                 10.0           RET                      ;EXIT-CLOCK CYCLE

                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*SUBROUTINE COMPARE
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*COMPPE -- ROUTINE TO COMPLIMENT THE EXPECTED PARITY BIT
                                        ;*      IN 'PARBEXP' STORAGE.
                                        S
                                        ;*********************************************************************

4579  3A  6A  46          13.0   COMPPE: LDA     PARBEXP          ;GET THE CURRENT PARITY BIT INFO
457C  A7                  4.0           ANA     A
457D  CA  85  45          10.0           JZ      PEBTON           ;SET IT ON

4580  AF                  4.0   PEBTOFF: XRA    A
4581  32  6A  46          13.0           STA     PARBEXP
4584  C9                 10.0           RET

4585  3E  04              7.0   PEBTON: MVI     A,4
4587  32  6A  46          13.0           STA     PARBEXP          ;SET IT ON
458A  C9                 10.0           RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*      SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*      CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*      CLEAR PORT SELECT FOR TRANSPORT
                                        ;*      CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*      CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*      CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************
458B                                    CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
458C                                            PUSH    B
458D 06 00                                      MVI     B,0             ;START TO CLEAR AT PORT #0
458F DB E0                              CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4591 E6 80                                      ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4593 B0                                         ORA     B               ;ADD IN THE SELECTED PORT #
4594 D3 E0                                      OUT     MBSEL           ;RESET TO THIS PORT
4596 3E 80                                      MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4598 D3 40                                      OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
459A AF                                         XRA     A               ;CLEAR TU COMMAND A
459B D3 40                                      OUT     TCMD            ;
459D 3E 81                                      MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
459F D3 40                                      OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
45A1 3E 00                                      MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
45A3 D3 40                                      OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
45A5 AF                                         XRA     A
45A6 D3 44                                      OUT     TAMT            ;CLEAR AMTIE WORD
45A8 D3 48                                      OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
45AA D3 4C                                      OUT     PENAB           ;CLEAR PORT ENABLE WORD
45AC 04                                         INR     B               ;POINT TO THE NEXT PORT TO CLEAR
45AD 78                                         MOV     A,B
45AE FE 04                                      CPI     4               ;DONE?
45B0 C2 8F 45                                   JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
45B3 C1                                         POP     B               ;RESET B & C
45B4 F1                                         POP     PSW             ;ALL DONE
45B5 C9                                         RET                     ;EXIT
```

```
                                .SBTTL  TABLE-EXPECTED XMC OUTPUT PC
45B6                            PXPTBL:                         ;CHARACTER 1 - ALL ZEROS
45B6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45B7  00                                .BYTE   $00             ;CHARACTER 2 - ALL ZEROS
45B8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45B9  00                                .BYTE   $00             ;CHARACTER 3 - ALL ZEROS
45BA  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45BB  00                                .BYTE   $00             ;CHARACTER 4 - ALL ZEROS
45BC  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45BD  00                                .BYTE   $00             ;CHARACTER 5 - ALL ZEROS
45BE  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45BF  00                                .BYTE   $00             ;CHARACTER 6 - ALL ZEROS
45C0  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C1  00                                .BYTE   $00             ;CHARACTER 7 - ALL ZEROS
45C2  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C3  00                                .BYTE   $00             ;CHARACTER 8 - ALL ZEROS
45C4  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C5  00                                .BYTE   $00             ;CHARACTER 9 - ALL ZEROS
45C6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C7  00                                .BYTE   $00             ;CHARACTER 10 - ALL ZEROS
45C8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C9  00                                .BYTE   $00             ;CHARACTER 11 - ALL ZEROS
45CA  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45CB  00                                .BYTE   $00             ;CHARACTER 12 - ALL ZEROS
45CC  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45CD  00                                .BYTE   $00             ;CHARACTER 13 - ALL ZEROS
45CE  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45CF  00                                .BYTE   $00             ;CHARACTER 14 - ALL ZEROS
45D0  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D1  00                                .BYTE   $00             ;CHARACTER 15 - ALL ZEROS
45D2  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D3  00                                .BYTE   $00             ;CHARACTER 16 - ALL ZEROS
45D4  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D5  00                                .BYTE   $00             ;CHARACTER 17 - ALL ZEROS
45D6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D7  00                                .BYTE   $00             ;CHARACTER 18 - ALL ZEROS
45D8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D9  00                                .BYTE   $00             ;CHARACTER 19 - ALL ZEROS
45DA  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45DB  00                                .BYTE   $00             ;CHARACTER 20 - ALL ZEROS
45DC  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45DD  00                                .BYTE   $00             ;CHARACTER 21 - ALL ZEROS
45DE  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45DF  00                                .BYTE   $00             ;CHARACTER 22 - ALL ZEROS
45E0  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E1  00                                .BYTE   $00             ;CHARACTER 23 - ALL ZEROS
45E2  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E3  00                                .BYTE   $00             ;CHARACTER 24 - ALL ZEROS
45E4  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E5  00                                .BYTE   $00             ;CHARACTER 25 - ALL ZEROS
45E6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E7  00                                .BYTE   $00             ;CHARACTER 26 - ALL ZEROS
45E8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E9  00                                .BYTE   $00             ;CHARACTER 27 - ALL ZEROS
```

```
                                                J 3
 1972  45EA  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1973  45EB  00                    .BYTE   $00             ;CHARACTER 28 - ALL ZEROS
 1974  45EC  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1975  45ED  00                    .BYTE   $00             ;CHARACTER 29 - ALL ZEROS
 1976  45EE  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1977  45EF  00                    .BYTE   $00             ;CHARACTER 30 - ALL ZEROS
 1978  45F0  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1979  45F1  00                    .BYTE   $00             ;CHARACTER 31 - ALL ZEROS
 1980  45F2  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1981  45F3  00                    .BYTE   $00             ;CHARACTER 32 - ALL ZEROS
 1982  45F4  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1983  45F5  00                    .BYTE   $00             ;CHARACTER 33 - ALL ZEROS
 1984  45F6  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1985  45F7  00                    .BYTE   $00             ;CHARACTER 34 - ALL ZEROS
 1986  45F8  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1987  45F9  00                    .BYTE   $00             ;CHARACTER 35 - ALL ZEROS
 1988  45FA  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1989  45FB  00                    .BYTE   $00             ;CHARACTER 36 - ALL ZEROS
 1990  45FC  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1991  45FD  00                    .BYTE   $00             ;CHARACTER 37 - ALL ZEROS
 1992  45FE  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1993  45FF  00                    .BYTE   $00             ;CHARACTER 38 - ALL ZEROS
 1994  4600  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1995  4601  00                    .BYTE   $00             ;CHARACTER 39 - ALL ZEROS
 1996  4602  FF                    .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
 1997  4603  00                    .BYTE   $00             ;CHARACTER 40 - ALL ZEROS
 1998  4604  FF                    .BYTE   $FF
 1999
 2000                          ;HERE IS THE END OF PREAMBLE...
 2001
 2002  4605  FF                            .BYTE   $FF             ;CHARACTER 41 - ALL ONES (NO WCLK)
 2003  4606  00                            .BYTE   $00             ;2ND HALF PE BIT CELL TIME
 2004                                                                ;
 2005                          ;6 BYTE DATA GROUP....DATA = 0'S & 1'S
 2006                                                                ;
 2007  4607  00                            .BYTE   $00             ;DATA BYTE 1
 2008  4608  FF                            .BYTE   $FF             ;BYTE A 11-NORMAL = 000
 2009
 2010  4609  FF                            .BYTE   $FF             ;DATA BYTE 2
 2011  460A  00                            .BYTE   $00             ;BYTE B 11-NORMAL = 377
 2012
 2013  460B  00                            .BYTE   $00             ;DATA BYTE 3
 2014  460C  FF                            .BYTE   $FF             ;DATA = 000
 2015
 2016  460D  FF                            .BYTE   $FF             ;DATA BYTE 4
 2017  460E  00                            .BYTE   $00             ;DATA = 377
 2018
 2019  460F  00                            .BYTE   $00             ;DATA BYTE 5
 2020  4610  FF                            .BYTE   $FF             ;DATA = 000
 2021
 2022  4611  FF                            .BYTE   $FF             ;DATA BYTE 6
 2023  4612  00                            .BYTE   $00             ;DATA = 000 ... LAST DATA BYTE
 2024
 2025                          ;XMC SHOULD SEND 6 DATA BYTES THEN 2 PAD BYTES (000)
```

```
2026                                    ;PADS ARE IGNORED BY THE TRANSLATOR...
2027                                                    ;
2028  4613  FF                          .BYTE   $FF       ;POSTAMBLE GROUP
2029  4614  00                          .BYTE   $00       ;CHARACTER 1 - ALL ONES
2030  4615  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2031  4616  FF                          .BYTE   $FF       ;CHARACTER 2 - ALL ZEROS
2032  4617  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2033  4618  FF                          .BYTE   $FF       ;CHARACTER 3 - ALL ZEROS
2034  4619  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2035  461A  FF                          .BYTE   $FF       ;CHARACTER 4 - ALL ZEROS
2036  461B  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2037  461C  FF                          .BYTE   $FF       ;CHARACTER 5 - ALL ZEROS
2038  461D  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2039  461E  FF                          .BYTE   $FF       ;CHARACTER 6 - ALL ZEROS
2040  461F  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2041  4620  FF                          .BYTE   $FF       ;CHARACTER 7 - ALL ZEROS
2042  4621  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2043  4622  FF                          .BYTE   $FF       ;CHARACTER 8 - ALL ZEROS
2044  4623  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2045  4624  FF                          .BYTE   $FF       ;CHARACTER 9 - ALL ZEROS
2046  4625  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2047  4626  FF                          .BYTE   $FF       ;CHARACTER 10 - ALL ZEROS
2048  4627  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2049  4628  FF                          .BYTE   $FF       ;CHARACTER 11 - ALL ZEROS
2050  4629  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2051  462A  FF                          .BYTE   $FF       ;CHARACTER 12 - ALL ZEROS
2052  462B  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2053  462C  FF                          .BYTE   $FF       ;CHARACTER 13 - ALL ZEROS
2054  462D  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2055  462E  FF                          .BYTE   $FF       ;CHARACTER 14 - ALL ZEROS
2056  462F  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2057  4630  FF                          .BYTE   $FF       ;CHARACTER 15 - ALL ZEROS
2058  4631  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2059  4632  FF                          .BYTE   $FF       ;CHARACTER 16 - ALL ZEROS
2060  4633  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2061  4634  FF                          .BYTE   $FF       ;CHARACTER 17 - ALL ZEROS
2062  4635  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2063  4636  FF                          .BYTE   $FF       ;CHARACTER 18 - ALL ZEROS
2064  4637  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2065  4638  FF                          .BYTE   $FF       ;CHARACTER 19 - ALL ZEROS
2066  4639  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2067  463A  FF                          .BYTE   $FF       ;CHARACTER 20 - ALL ZEROS
2068  463B  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2069  463C  FF                          .BYTE   $FF       ;CHARACTER 21 - ALL ZEROS
2070  463D  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2071  463E  FF                          .BYTE   $FF       ;CHARACTER 22 - ALL ZEROS
2072  463F  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2073  4640  FF                          .BYTE   $FF       ;CHARACTER 23 - ALL ZEROS
2074  4641  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2075  4642  FF                          .BYTE   $FF       ;CHARACTER 24 - ALL ZEROS
2076  4643  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2077  4644  FF                          .BYTE   $FF       ;CHARACTER 25 - ALL ZEROS
2078  4645  00                          .BYTE   $00       ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2079  4646  FF                          .BYTE   $FF       ;CHARACTER 26 - ALL ZEROS
```

```
2080  4647  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2081  4648  FF                .BYTE   $FF          ;CHARACTER 27 - ALL ZEROS
2082  4649  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2083  464A  FF                .BYTE   $FF          ;CHARACTER 28 - ALL ZEROS
2084  464B  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2085  464C  FF                .BYTE   $FF          ;CHARACTER 29 - ALL ZEROS
2086  464D  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2087  464E  FF                .BYTE   $FF          ;CHARACTER 30 - ALL ZEROS
2088  464F  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2089  4650  FF                .BYTE   $FF          ;CHARACTER 31 - ALL ZEROS
2090  4651  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2091  4652  FF                .BYTE   $FF          ;CHARACTER 32 - ALL ZEROS
2092  4653  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2093  4654  FF                .BYTE   $FF          ;CHARACTER 33 - ALL ZEROS
2094  4655  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2095  4656  FF                .BYTE   $FF          ;CHARACTER 34 - ALL ZEROS
2096  4657  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2097  4658  FF                .BYTE   $FF          ;CHARACTER 35 - ALL ZEROS
2098  4659  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2099  465A  FF                .BYTE   $FF          ;CHARACTER 36 - ALL ZEROS
2100  465B  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2101  465C  FF                .BYTE   $FF          ;CHARACTER 37 - ALL ZEROS
2102  465D  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2103  465E  FF                .BYTE   $FF          ;CHARACTER 38 - ALL ZEROS
2104  465F  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2105  4660  FF                .BYTE   $FF          ;CHARACTER 39 - ALL ZEROS
2106  4661  00                .BYTE   $00          ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2107  4662  FF                .BYTE   $FF          ;CHARACTER 40 - ALL ZEROS
2108  4663  00                .BYTE   $00          ;1ST HALF
2109  4664  FF                .BYTE   $FF          ;CHARACTER 41 - ALL ZEROS
2110  4665  00                .BYTE   $00          ;THIS IS THE LAST TICK CHECK
2111  4666              PTBLEN:
2112
2113
2114
2115                    .SBTTL  PROGRAM VARIABLES
2116
2117  4666  00          UNITMP: .BYTE   0           ;UNIT MAP
2118  4667  00  00      POINTER: .WORD  0           ;POINTER TO EXPECTED DATA TABLE
2119  4669  00          PNTCNT: .BYTE   0           ;COUNT OF COMPARE POINTER
2120  466A  00          PARBEXP: .BYTE  0           ;STORAGE FOR EXPECTED PARITY BIT
2121        0000                .END
```

```
A      =%0007          ABOCN0  43EF  G       ADATA  = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098           ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000          BADST  = 0090          BGNPRE  43F8
BIT0   = 0001          BIT1   = 0002          BIT15  = 8000          BIT2   = 0004
BIT3   = 0008          BIT4   = 0010          BIT5   = 0020          BIT6   = 0040
BIT7   = 0080          BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A
BRKRAM= 4F10           BRKSTR= 4E60           BRKXCT= 4F00           BSAVE   4F9C
BYTCNT= 00D4           BYTEH   4F24           BYTEL   4F23           C      =%0001
CASCT   4F21           CASCTL= 00A0           CASSTA= 00A0           CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1           CBYTH  = 008B          CBYTL  = 008A
CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C
CHPTIE= 0028           CHOTIE= 0020           CH1TIE= 0021           CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024           CH5TIE= 0025           CH6TIE= 0026
CH7TIE= 0027           CKLOP  = 2815          CLEAR   4588          CLK     43DC
CLKCTL= 00F0           CLKSYS  456E           CLK1    43DA          CLOCK   4F26
CLRLP   458F           CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B
CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F
CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7
COMPPE  4579           CRCWRD= 0018           CSAVE   4F9D          CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080           C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040           C.WCS  = 0002
D      =%0002          DATACT= 00D0           DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010           DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001           DDRCO  = 0088
DDRCTL= 00DB           DIAFLG  4F22           DIAGPG= 4300           DIAGRM= 4F90
DIARD  = 0008          DONE1  = 0045          DONINT= 0010           DSAVE   4F9E
DSE    = 0006          DUMMY   4355  G        D.ATH0= 0001           D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003           ERFLG   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F           EXIT    456B          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001           E.TTEC= 0002           E.UNC  = 0004
FIFORD= 006A           FORMAT  4F25           FOUND   435E          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002           GOODTM= 0092           H      =%0004
HLSAVE  4FA0           IE     = 0008          INTSTA= 00E0           ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020
I7.5   = 0040          KCALL  = 005F          KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8           KEY1   = 0078
KEY10  = 006D          KEY11  = 006E          KEY12  = 006F          KEY13  = 005C
KEY14  = 005D          KEY15  = 005E          KEY16  = 005F          KEY17  = 003C
KEY18  = 003D          KEY19  = 003E          KEY2   = 0079          KEY20  = 003F
KEY3   = 007A          KEY4   = 007B          KEY5   = 0074          KEY6   = 0075
KEY7   = 0076          KEY8   = 0077          KEY9   = 006C          KINTA  = 006F
KLDAD  = 003D          KN0    = 003C          KN1    = 005C          KN2    = 005D
KN3    = 005E          KN4    = 006C          KN5    = 006D          KN6    = 006E
```

```
KN7    = 0074          KN8    = 0075          KN9    = 0076          KU2    = 0079
KU3    = 007A          KU8    = 0077          L      =%0005          LBLANK= 000F
LCE    = 0008          LCH    = 000C          LCL    = 000D          LCP    = 000E
LC0    = 0000          LC1    = 0001          LC2    = 0002          LC3    = 0003
LC4    = 0004          LC5    = 0005          LC6    = 0006          LC7    = 0007
LC8    = 0008          LC9    = 0009          LDLEDA= 00CA          LDLEDB= 00CB
LDLEDC= 00CC          LDLEDD= 00CD          LDLEDE= 00CE          LDLEDF= 00CF
LKDIAG= 2800          LKKBD  = 004C          LKKEY  = 0049          LKLWMG= 0058
LKLWMP= 0055          LKLWPG= 0052          LKLWPP= 004F          LKMOD7= 0046
LKOPR  = 0046          LPFLG   4F94          LPNUM   4F92          M      =%0006
MBSEL  = 00E0          MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF
MINUS  = 000A          MB.    = 8000          MSE    = 0008          MSGN   = 000F
MTACLR= 0000          MT.AKA= 0020          MT.CPE= 0080          MT.DSE= 0001
MT.FWD= 0040          MT.INH= 0008          MT.LWR= 0004          MT.MOT= 0002
MT.NWT= 0080          MT.PEC= 0040          MT.PSB= 0004          MT.PS0= 0001
MT.PS1= 0002          MT.REV= 0020          MT.WRT= 0010          MT.Z   = 0008
M.ATA  = 0080          M.CAPE= 0020          M.CONT= 0080          M.DEM  = 0020
M.EBL  = 0010          M.EXC  = 0008          M.FAIL= 0008          M.ILR  = 0010
M.INIT= 0010          M.OCC  = 0020          M.ONLI= 0001          M.PE   = 0040
M.PORT= 0080          M.RDEN= 0002          M.RDPE= 0008          M.RUN  = 0004
M.SCLK= 0001          M.TRA  = 0040          M.UNIT= 0007          M.WCLK= 0040
M.WCLN= 0080          M.WREN= 0080          M5.5   = 0001          M6.5   = 0002
M7.5   = 0004          NOTCAP= 0088          OKAY   = 00FF          OPRRAM= 4300
OPSTRT= 0058          OPVER  = 0040          PADCNT= 00D5          PADCRC= 0080
PARBEX  466A          PDIAG  = 0048          PEBTOF  4580          PEBTON  4585
PEID   = 008A          PENAB  = 004C          PESET  = 0001          PL     = 00B1
PNTCNT  4669          POINTE  4667          PRDD   = 004C          PREC4   44D0  G
PREC1   440A  G       PREC1A  4429  G       PREC2   4448  G       PREC2A  445D  G
PREC3   4496  G       PREC3A  449F          PREC3B  44B4  G       PREC5   44EF  G
PREC5A  4504  G       PRENF  = 009C          PS     = 00B2          PSTAT  = 0048
PSW    =%0009          PTBLEN  4666          PXPTBL  45B6          P.AMTP= 0001
P.BCTC= 0040          P.CMDP= 0020          P.INTE= 0080          P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004          P.RPST= 0002
P.RP0E= 0020          P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010
P.SING= 0080          P.STAT= 0002          P.STPE= 0080          P.TACH= 0008
P.TUPR= 0010          P.WCSP= 0004          P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010          P.WP0E= 0008          P.WP1E= 0004          P.WP2E= 0008
P.WP3E= 0004          P.5VOK= 0002          QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047          RCHOK  = 0046          RCHTST= 000C          RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          RECCN1  453C  G
RECCN2  4565  G       RECCN3  4557  G       RECEND  4529          REND   = 0014
REQST  = 2806          RESCHR= 00D1          RESTCK  43BA          RESTOK  43D0  G
REVTST= 0064          REWIND= 0004          RFIFOL= 0008          RGCLK  = 0002
RGCRI  = 0003          RIBG   = 0001          RILL   = 0012          RINST  = 000C
RMCTST= 0008          RMK2   = 0013          RNGP   = 0000          RPATH  = 0001
RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003          RPCTL  = 0009
RPEI   = 0002          RPFAIL= 0000          RPF1   = 009D          RPF2   = 009E
RPOK   = 0043          RPOSTN= 0016          RPSTA  = 0015          RRCMT  = 000A
RSTAT  = 0002          RTIEB  = 000A          RTIER  = 0030          RTM    = 0005
RUNKI  = 0009          RUPTST= 005E          RWDUNL= 0005          R.AMT  = 0001
R.BOP  = 0008          R.DATA= 0040          R.DON  = 0002          R.DRDY= 0010
R.END  = 0010          R.ILL  = 0004          R.JVOK= 0004          R.MK2  = 0008
```

XMC2.M80 SYMBOL TABLE

```
R.PLOD= 0008    R.PL00= 0010    R.PL01= 0020    R.POST= 0020
R.STNM= 0002    R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080
R.TSTD= 0040    R.VOK = 0080    R00H  = 0081    R00L  = 0080
R01H  = 0083    R01L  = 0082    R02H  = 0085    R02L  = 0084
R03H  = 0087    R03L  = 0086    R04H  = 0089    R04L  = 0088
R05H  = 008B    R05L  = 008A    R06H  = 008D    R06L  = 008C
R07H  = 008F    R07L  = 008E    R10H  = 0091    R10L  = 0090
R11H  = 0093    R11L  = 0092    R12H  = 0095    R12L  = 0094
R13H  = 0097    R13L  = 0096    R14H  = 0099    R14L  = 0098
R15H  = 009B    R15L  = 009A    R16H  = 009D    R16L  = 009C
R17H  = 009F    R17L  = 009E    R7.5  = 0010    SELCLR= 0000
SEQERR  4560    SETATA= 00A1    SID   = 0080    SOD   = 0080
SOE   = 0040    SP    =%0008    SSCLK = 0040    SSTEP = 0005
STACK = 4FFF    STATRM= 4F20    STPCT   4F20    STRSP = 5000
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TMF   = 0099    TMRDY = 0040
TRKENA= 00D2    TSET  = 2803    TSTEND= 2818    TSTS  = 0040
TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010    T.ATH0= 0001
T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002    T.FPT = 0001
T.NTHR= 0004    T.ONL = 0020    T.PES = 0008    T.PSBJ= 0020
T.PS0J= 0008    T.PS1J= 0010    T.RDY = 0080    T.RDY0= 0040
T.RWD = 0010    T.SCLK= 0002    T1LOOP  436C    UIBG  = 00A1
UNITMP  4666    VALFC   4F98    VALTB   4F95    VELTST= 005B
WDR.P = 0010    WMCCTL= 00D3    WMCERR= 00DA    WMCSTA= 00D0
WRTCLK= 0000    WRTDAT= 00D3    W.ACRC= 0004    W.CRC = 0008
W.DIAG= 0002    W.DONN= 0040    W.ECC = 0010    W.ENAB= 0080
W.ERR = 0020    W.FMT = 0070    W.GCR = 0010    W.LEFT= 0004
W.ONES= 0020    W.RESI= 0002    W.REV = 0004    W.ROME= 0010
W.RST = 0001    W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020
X     =%000A    X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002
X.ROME= 0001    X.WCLK= 0001    Y     =%000B    .     = 466B
```

ERRORS DETECTED: 0

*XMC2.A78/PTP,XMC2=NLIST,PARAM,MACRO,LIST,XMC2
RUN-TIME: 4 5 0 SECONDS
CORE USED: 10K

```
                                .TITLE  XMC3 - TRANSLATOR MICRO CONTROLLER PART #3
                                .SBTTL  TEST 1 - XMC - GCR PREAMBLE/POSTAMBLE TEST
                                ;ID     XMC3-TRANSLATOR MICRO CONTROLLER PART #3
4300                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR PREAMBLE/POSTAMBLE TEST
4300                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A GCR RECORD OF SEVEN BYTES
                                ;AND THEN SINGLE STEPS THE WRITE PATH (WMC, XMC) WHILE MONITORING THE
                                ;XMC ''XL WCLK'' BIT.  THE OUTPUT OF THE TRANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, END MARK, DATA GROUP, MARK-2, RESIDUAL GROUP, CRC GROUP, AND POSTAMBLE.
                                ;THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT, DATA STATES AND FLUX REVERSAL RATE.
4300                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      INITIALIZE THE BYTE COUNTER, PAD COUNTER, AND ERROR COUNTER
                                ;*      CLEAR THE ERROR COUNTER
                                ;*      REQUEST A TU PORT FROM THE HOST
                                ;*      CALL SUBROUTINE CLEAR
                                ;*      IF NO UNITS SELECTED
                                ;*      :       THEN-ERROR-ABORT TEST
                                ;*      :       ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      SET LOOP COMMAND/STATUS ON SELECTED PORT
                                ;*      CLEAR THE COMMAND BYTE FOR SELECTED PORT
                                ;*      SET WRITE PATH ENABLE
                                ;*      INITIALIZE THE EXPECTED RESULTS TABLE POINTER
                                ;*      SET WMC/XMC RESTART
                                ;*      SELECT THE SINGLE STEP CLOCK
                                ;*      CLEAR WMC/XMC RESTART
                                ;*      LOAD THE BYTE COUNT WITH 8 (BITS 7:0)
                                ;*      CLEAR THE BYTE COUNT (BITS 15:8)
                                ;*      SET DDR CONTROL TO ''IN''
                                ;*      SELECT THE IMAGE MODE OF DATA FORMATTING
                                ;*      LOAD THE RESIDUAL CHARACTER WITH 6
                                ;*      LOAD THE PAD COUNT WITH 6 (BITS 7:0)
                                ;*      CLEAR THE PAD COUNT (BITS 15:8)
                                ;*      ENABLE ALL TRACKS LOAD TRACK ENABLE WITH ONES
                                ;*      SET THE CLOCK COUNT TO 180(10)
                                ;*      BGNDO
                                ;*      :       CALL SUBROUTINE CLKSYS
                                ;*      :       DECREMENT THE CLOCK COUNT
                                ;*      :       DO UNTIL CLOCK COUNT=0
                                ;*      ENDDO
                                ;*      SET WRITE MICRO TO WRITE MICRO ENABLE, TRANSLATOR ENABLE, GCR, AND WRITE
                                ;*      SET THE CLOCK COUNT TO 256(10)
                                ;*      BGNDO
```

```
1374                                    ;*    :    CALL SUBROUTINE CLKSYS
1375                                    ;*    :    DECREMENT THE CLOCK COUNT
1376                                    ;*    :    DO UNTIL CLOCK COUNT=0 OR XL WRITE CLOCK=1
1377                                    ;*    ENDDO
1378                                    ;*    IF CLOCK COUNT=0
1379                                    ;*    :    THEN-FATAL ERROR-ABORT TEST
1380                                    ;*    :    ELSE-CONTINUE
1381                                    ;*    ENDIF
1382                                    ;*    BGNDO
1383                                    ;*    :    CALL SUBROUTINE CLKSYS
1384                                    ;*    :    CALL SUBROUTINE CLKSYS
1385                                    ;*    :    INPUT THE TRANSLATOR OUTPUT DATA
1386                                    ;*    :    IF THE ACTUAL TRANSLATOR OUTPUT=EXPECTED TRANSLATOR OUTPUT
1387                                    ;*    :    :    THEN-CONTINUE
1388                                    ;*    :    :    ELSE-ERROR
1389                                    ;*    :    ENDIF
1390                                    ;*    :    INPUT THE TRANSLATOR OUTPUT DATA PARITY
1391                                    ;*    :    IF DATA OUTPUT DURING DATA GROUP
1392                                    ;*    :    :    THEN-EXPECTED PARITY=ODD PARITY FOR OUTPUT DATA
1393                                    ;*    :    :    ELSE-EXPECTED PARITY=1
1394                                    ;*    :    ENDIF
1395                                    ;*    :    IF INPUT PARITY=EXPECTED PARITY
1396                                    ;*    :    :    THEN-CONTINUE
1397                                    ;*    :    :    ELSE-ERROR
1398                                    ;*    :    ENDIF
1399                                    ;*    :    SET THE LOOP COUNT TO 14(8)
1400                                    ;*    :    BGNDO
1401                                    ;*    :    :    CALL SUBROUTINE CLKSYS
1402                                    ;*    :    :    DECREMENT THE LOOP COUNT
1403                                    ;*    :    :    DO UNTIL THE LOOP COUNT = 0
1404                                    ;*    :    ENDDO
1405                                    ;*    :    IF XL WRITE CLOCK = 1
1406                                    ;*    :    :    THEN - CONTINUE
1407                                    ;*    :    :    ELSE - ERROR - IMPROPER TRANSITION RATE FOR GCR
1408                                    ;*    :    ENDIF
1409                                    ;*    :    INCREMENT THE EXPECTED RESULTS TABLE POINTER
1410                                    ;*    :    DO UNTIL THE EXPECTED RESULTS POINTER=END OF TABLE
1411                                    ;*    ENDDO
1412                                    ;*    SET THE CLOCK COUNT TO 256(10)
1413                                    ;*    BGNDO
1414                                    ;*    :    CALL SUBROUTINE CLKSYS
1415                                    ;*    :    DECREMENT THE CLOCK COUNT
1416                                    ;*    :    DO UNTIL CLOCK COUNT=0 OR TRANSLATOR OR WRITE MICRO DONE=0
1417                                    ;*    ENDDO
1418                                    ;*    IF CLOCK COUNT=0
1419                                    ;*    :    THEN-ERROR-TIMEOUT WAITING FOR WRITE MICRO DONE
1420                                    ;*    :    ELSE-CONTINUE
1421                                    ;*    ENDIF
1422                                    ;*    IF TRANSLATOR DONE
1423                                    ;*    :    THEN-ERROR-TRANSLATOR DONE BEFORE WRITE MICRO
1424                                    ;*    :    ELSE-SET THE CLOCK COUNT TO 256(10)
1425                                    ;*    :    BGNDO
1426                                    ;*    :    :    CALL SUBROUTINE CLKSYS
1427                                    ;*    :    :    DECREMENT THE CLOCK COUNT
```

```
1428                                    ;*     :     :    DO UNTIL TRANSLATOR DONE OR CLOCK COUNT=0
1429                                    ;*     :     ENDDO
1430                                    ;*     :     IF CLOCK COUNT=0
1431                                    ;*     :     :    THEN-ERROR-TIMEOUT WAITING FOR TRANSLATOR DONE
1432                                    ;*     :     :    ELSE-CONTINUE
1433                                    ;*     :     ENDIF
1434                                    ;*     ENDIF
1435                                    ;*ENDTST
1436  4300                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1437                                    ;*XMC3 MICRO TEST 01
1438                                    ;*XMC3 MICRO ERROR 01
1439                                    ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
1440                                    ;*M8958, M8959, M8955'S
1441                                    ;*OPERATOR ERROR NO TM78 UNITS SELECTED
1442                                    ;*FATAL ERROR - TEST ABORTED
1443                                    ;*
1444                                    ;*XMC3 MICRO TEST 01
1445                                    ;*XMC3 MICRO ERROR 02
1446                                    ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
1447                                    ;*M8958, M8959, M8955'S
1448                                    ;*WRITE PATH TIMEOUT WHILE WAITING FOR 'XL CLK'' TO SET
1449                                    ;*FATAL ERROR - TEST ABORTED
1450                                    ;*
1451                                    ;*XMC3 MICRO TEST 01
1452                                    ;*XMC3 MICRO ERROR 03
1453                                    ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
1454                                    ;*M8958, M8959, M8955'S
1455                                    ;*DATA STATE RECEIVED FROM THE XMC INCORRECT
1456                                    ;*ACTUAL = NNNN
1457                                    ;*EXPECTED = NNNN
1458                                    ;*TRANSITION COUNT = LLL
1459                                    ;*
1460                                    ;*XMC3 MICRO TEST 01
1461                                    ;*XMC3 MICRO ERROR 04
1462                                    ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
1463                                    ;*M8958, M8959, M8955'S
1464                                    ;*PARITY STATE RECEIVED FROM THE XMC INCORRECT
1465                                    ;*ACTUAL = NNNN
1466                                    ;*EXPECTED = NNNN
1467                                    ;*TRANSITION COUNT = LLL
1468                                    ;*
1469                                    ;*XMC3 MICRO TEST 01
1470                                    ;*XMC3 MICRO ERROR 05
1471                                    ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
1472                                    ;*M8958, M8959, M8955'S
1473                                    ;*TRANSLATOR 'XL WCLK'' NOT SET AFTER EIGHT CLOCK CYCLES
1474                                    ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1475                                    ;*TRANSITION COUNT = LLL
1476                                    ;*
1477                                    ;*XMC3 MICRO TEST 01
1478                                    ;*XMC3 MICRO ERROR 06
```

```
                                        ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                        ;*M8958, M8959, M8955'S
                                        ;*'WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
                                        ;*
                                        ;*XMC3 MICRO TEST 01
                                        ;*XMC3 MICRO ERROR 07
                                        ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                        ;*M8958, M8959, M8955'S
                                        ;*'XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
                                        ;*
                                        ;*XMC3 MICRO TEST 01
                                        ;*XMC3 MICRO ERROR 10
                                        ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                        ;*M8958, M8959, M8955'S
                                        ;*'XMC NOT DONE'' RESET BEFORE 'WMC NOT DONE'' RESET
1494  4300                              $
 (1)                                    ; ***************************************************************************
1495
1496  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01            7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1497                                    ;%XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
1498                                    ;&M8958, M8959, M8955'S
1499
1500  4305  AF                 4.0     T1LOOP: XRA     A               ;CLEAR UNIT MAP STORAGE
1501  4306  32  91  46        13.0             STA     EXDLPC          ;CLEAR THE ERROR COUNTER
1502  4309  3E  30             7.0             MVI     A,@60           ;INIT THE BYTE COUNTER
1503  430B  D3  D7            10.0             OUT     CNTCTL          ;TO NORMAL MODE
1504  430D  3E  70             7.0             MVI     A,@160          ;INIT THE PAD COUNTER
1505  430F  D3  D7            10.0             OUT     CNTCTL          ;TO NORMAL MODE
1506  4311  3E  B0             7.0             MVI     A,@260          ;INIT THE ERROR COUNTER
1507  4313  D3  D7            10.0             OUT     CNTCTL          ;TO NORMAL MODE
1508  4315                              CLRECT
 (1)  4315  AF                 4.0                     XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4316  D3  D6            10.0                     OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4318  D3  D6            10.0                     OUT     ERRCNT          ;CLEAR BITS 15-8
1509
1510                                    ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
1511                                    ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.
1512
1513  431A                              REQ     @7,0,0,0
 (1)  431A  CD  06  28       18.0                 CALL    REQST
 (1)  431D  00                                    .BYTE   0               ;DATA PATTERN NUMBER
 (1)  431E  00  00                                .WORD   0               ;SYSTEM '0'' COUNT
 (1)  4320  00  00                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4322  00                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  4323  07                                    .BYTE   @7              ;REQUEST CODE
1514  4324                              RIN     R12L
 (1)  4324  DB  94            10.0                 IN      R12L            ;READ R12L INTO AC
 (1)  4326  7F                 4.0                 MOV     A,A             ;RETRY LINK
1515  4327  32  8D  46        13.0             STA     UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
1516  432A  AF                 4.0             XRA     A
1517  432B  CD  6C  45        18.0             CALL    CLEAR
1518  432E  06  00             7.0             MVI     B,0
```

```
1519  4330  3A 8D 46   13.0  T1LPA:  LDA   UNITMP          ;GET SELECTED UNIT MAP
1520  4333  E6 01       7.0         ANI   $01             ;UNIT 0 PRESENT?
1521  4335  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1522  4338  04          4.0         INR   B
1523  4339  3A 8D 46   13.0         LDA   UNITMP          ;
1524  433C  E6 02       7.0         ANI   $02             ;UNIT 1 PRESENT?
1525  433E  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1526  4341  04          4.0         INR   B
1527  4342  3A 8D 46   13.0         LDA   UNITMP          ;
1528  4345  E6 04       7.0         ANI   $04             ;UNIT 2 PRESENT?
1529  4347  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1530  434A  04          4.0         INR   B
1531  434B  3A 8D 46   13.0         LDA   UNITMP
1532  434E  E6 08       7.0         ANI   $08             ;UNIT 3 PRESENT
1533  4350  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1534                                                      ;ELSE - OPERATOR ERROR
1535  4353                          ERR   EXIT,DUMMY
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4353  CD 09 28   18.0                CALL  ERLP          ;PROCESS ERROR - DO 2.3
 (1)       0001                    MSGN  =     MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4356  01                     .BYTE MSGN          ;MESSAGE NUMBER ID
 (1)  4357  00                     .BYTE
 (1)  4358  CD 15 28   18.0  DUMMY:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  435B  DA 5D 45   10.0         JC    EXIT           ;LOOP ADDRESS IF LOOP SPECIFIED
1536                          ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1537                          ;>FATAL ERROR - TEST ABORTED
1538  435E  C3 5D 45   10.0         JMP   EXIT
1539
1540                          ;SET UP FOR THE DATA TRANSFER
1541  4361  DB E0       10.0  FOUND:  IN    INTSTA
1542  4363  E6 80        7.0         ANI   BIT7
1543  4365  B0          4.0         ORA   B
1544  4366  D3 E0      10.0         OUT   MBSEL
1545  4368  3E 40       7.0         MVI   A,P.LCS         ;KILL THE MTA BOARD
1546  436A  D3 48      10.0         OUT   PDIAG
1547  436C  AF          4.0         XRA   A
1548  436D  D3 40      10.0         OUT   TCMD
1549  436F  3E 10       7.0         MVI   A,P.WPEN        ;SET WRITE PATH ENABLE
1550  4371  D3 4C      10.0         OUT   PENAB
1551  4373  21 97 45   10.0         LXI   H,GXPTBL        ;LOAD ADDRESS OF EXPECTED
1552  4376  22 92 46   16.0         SHLD  POINTER         ;RESULTS TABLE INTO POINTER
1553  4379  3E 01       7.0         MVI   A,W.RST         ;LOAD THE WMC/XMC RESTART CODE
1554  437B  D3 D3      10.0         OUT   WMCCTL          ;ISSUE THE COMMAND
1555  437D  3E 05       7.0         MVI   A,SSTEP         ;SELECT SINGLE STEP WRITE CLOCK
1556  437F  D3 F0      10.0         OUT   CLKCTL          ;INITIALIZE THE CLOCK
1557  4381  32 96 46   13.0         STA   CCTLWD          ;SAVE SOFTWARE IMAGE
1558  4384  CD 60 45   18.0         CALL  CLKSYS          ;CLOCK THE SYSTEM TO ENABLE
1559  4387  CD 60 45   18.0         CALL  CLKSYS          ;RESTART TO TAKE HOLD
1560  438A  AF          4.0         XRA   A               ;CLEAR THE ACC.
1561  438B  D3 D3      10.0         OUT   WMCCTL          ;CLEAR THE WMC/XMC RESTART BIT
1562  438D  3E 07       7.0         MVI   A,7             ;GET THE BYTE COUNT - 1 DATA GROUP
1563  438F  3C          4.0         INR   A               ;CORRECT FOR INTEL 8253 CHIP
1564  4390  D3 D4      10.0         OUT   BYTCNT          ;LOAD BITS 7-0
1565  4392  AF          4.0         XRA   A
```

```
1566  4393  D3  D4        10.0          OUT     BYTCNT          ;CLEAR THE BYTE COUNTER BITS 15-8
1567  4395  32  94  46    13.0          STA     TRANSCNT        ;INITIALIZE THE TRANSITION COUNT
1568  4398  32  95  46    13.0          STA     TRANSCNT+1
1569  439B  32  8F  46    13.0          STA     DPCNT           ;CLEAR THE PARITY TEST COUNTER
1570  439E  3E  88         7.0           MVI     A,@210
1571  43A0  D3  DB        10.0          OUT     DDRCTL          ;SET THE DDR CONTROL TO 'IN'
1572  43A2  3E  50         7.0           MVI     A,$50           ;LOAD THE IMAGE MODE COMMAND
1573  43A4  D3  D0        10.0          OUT     DATACTL         ;SET THE DATA CONTROL MODE
1574  43A6  3E  06         7.0           MVI     A,6             ;LOAD THE RESIDUAL CHARACTER VALUE
1575  43A8  D3  D1        10.0          OUT     RESCHR          ;SET THE RESIDUAL CHARACTER
1576  43AA  3E  06         7.0           MVI     A,6             ;LOAD PAD COUNT VALUE
1577  43AC  3C             4.0           INR     A               ;CORRECT FOR INTEL 8253 CHIP
1578  43AD  D3  D5        10.0          OUT     PADCNT          ;WRITE TO WMC BITS 7-0
1579  43AF  AF             4.0           XRA     A               ;LOAD HI VALUE
1580  43B0  D3  D5        10.0          OUT     PADCNT          ;WRITE TO WMC
1581  43B2  D3  D2        10.0          OUT     TRKENA          ;1ST CLEAR THE TRACK ENABLE BITS
1582  43B4  D3  D2        10.0          OUT     TRKENA
1583  43B6  3E  FF         7.0           MVI     A,$FF           ;LOAD ACC WITH ALL ONES
1584  43B8  D3  D2        10.0          OUT     TRKENA          ;ENABLE PARITY TRACK
1585  43BA  D3  D2        10.0          OUT     TRKENA          ;ENABLE DATA TRACKS
1586  43BC  06  B4         7.0           MVI     B,180           ;LOAD THE CLOCK COUNT
1587
1588  43BE  CD  60  45     18.0  CLKRS:  CALL    CLKSYS          ;CLOCK IT
1589  43C1  05             4.0           DCR     B               ;
1590  43C2  C2  BE  43     10.0          JNZ     CLKRS           ;CONTINUE WITH RESTART CLOCKING
1591
1592  43C5  3E  D8         7.0           MVI     A,W.ENAB!X.ENAB!W.GCR!W.WRITE
1593  43C7  D3  D3        10.0          OUT     WMCCTL          ;START THE OPERATION
1594  43C9  06  FF         7.0  CLK1:   MVI     B,$FF           ;SET TIMEOUT COUNT TO 256
1595                                                            ;
1596  43CB  CD  60  45     18.0  CLK:    CALL    CLKSYS          ;SINGLE STEP THE WRITE PATH
1597                                                            ;
1598  43CE  DB  D0        10.0          IN      WMCSTA          ;GET THE WMC STATUS
1599  43D0  E6  01         7.0           ANI     X.WCLK          ;X.WCLK SET?
1600  43D2  C2  E7  43     10.0          JNZ     BGNPRE          ;YES-BEGINNING OF PREAMBLE
1601  43D5  05             4.0           DCR     B               ;NO-DECREMENT CLOCK
1602  43D6  C2  CB  43     10.0          JNZ     CLK             ;CONTINUE UNTIL ZERO
1603  43D9                                ERR     TSTEND,ABOCNO
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43D9  CD  09  28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43DC  02                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43DD  00                                 .BYTE
 (1)  43DE  CD  15  28     18.0          ABOCNO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E1  DA  18  28     10.0                  JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1604                            ;>WRITE PATH TIMEOUT WHILE WAITING FOR 'XL CLK' TO SET
1605                            ;>FATAL ERROR - TEST ABORTED
1606
1607  43E4  C3  18  28     10.0          JMP     TSTEND
1608
1609  43E7  CD  60  45     18.0  BGNPRE: CALL    CLKSYS          ;CLOCK THE SYSTEM
1610  43EA  CD  60  45     18.0          CALL    CLKSYS          ;CLOCK AGAIN
1611  43ED  DB  40        10.0          IN      TSTS            ;GET THE DATA READ
1612  43EF  47             4.0           MOV     B,A             ;SAVE IT IN B=ADATA
```

```
1613  43F0                                          ROUT    ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  43F0  D3  94          10.0                             OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43F2  7F               4.0                             MOV     A,A             ;RETRY LINK
1614                                                         ;
1615  43F3  2A  94  46       16.0                  LHLD    TRANSCNT        ;GET THE TRANSITION COUNT
1616  43F6  23                6.0                  INX     H               ;UPDATE
1617  43F7  22  94  46       16.0                  SHLD    TRANSCNT        ;RESTORE THE TRANSITION COUNT
1618  43FA  7D                4.0                  MOV     A,L             ;GET TRANSCOUNT LOW
1619  43FB                                         ROUT    R05L            ;WRITE TO CAS
 (1)  43FB  D3  8A          10.0                             OUT     R05L            ;WRITE AC INTO R05L
 (1)  43FD  7F               4.0                             MOV     A,A             ;RETRY LINK
1620  43FE  7C               4.0                  MOV     A,H             ;GET TRANSCOUNT HIGH
1621  43FF                                         ROUT    R05H            ;WRITE TO CAS
 (1)  43FF  D3  8B          10.0                             OUT     R05H            ;WRITE AC INTO R05H
 (1)  4401  7F               4.0                             MOV     A,A             ;RETRY LINK
1622  4402  2A  92  46       16.0                  LHLD    POINTER         ;GET POINTER TO EXPECTED DATA
1623  4405  7E                7.0                  MOV     A,M             ;GET EXPECTED DATA BYTE
1624  4406  32  8E  46       13.0                  STA     XDATA           ;SAVE FOR LATER
1625  4409  B8                4.0                  CMP     B               ;COMPARE WITH ACTUAL
1626  440A  CA  22  44        10.0                  JZ      PRECN1          ;CONTINUE IF EQUAL
1627                                                                       ;ELSE-ERROR
1628  440D                                         ROUT    EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  440D  D3  95          10.0                             OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  440F  7F               4.0                             MOV     A,A             ;RETRY LINK
1629  4410  3A  91  46       13.0  XDLPC1: LDA     EXDLPC          ;GET THE CURRENT COUNT AT ERROR
1630  4413  A7                4.0                  ANA     A
1631  4414  C2  1D  44        10.0                  JNZ     XDLPC2          ;STAY AT THIS COUNT IF NOT ZERO
1632  4417  3A  94  46       13.0                  LDA     TRANSCNT        ;GET THE COUNT AT THE ERROR
1633  441A  32  91  46       13.0                  STA     EXDLPC          ;SAVE IT
1634  441D                                XDLPC2: ERRB    XDLOOP,PRECN1,@4
 (1)                                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  441D  CD  12  28       18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0003                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4420  03                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4421  04                                          .BYTE   @4              ;PRINT ROUTINE NUMBER
 (1)  4422  CD  15  28       18.0                PRECN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4425  DA  2B  44       10.0                          JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1635                                         ;>DATA STATE RECEIVED FROM THE XMC INCORRECT
1636  4428  C3  46  44       10.0                  JMP     XDLPC           ;SKIP OVER ERROR LOOP CONTROL
1637
1638
1639
1640                                         ;HERE IS THE ERROR LOOP CONTROL
1641
1642  442B  3A  94  4F       13.0  XDLOOP: LDA     LPFLG           ;GET THE LOOP ON ERROR FLAG
1643  442E  A7                4.0                  ANA     A
1644  442F  C2  38  44        10.0                  JNZ     XDLP1           ;JUMP IF FLAG IS SET
1645  4432  32  91  46       13.0                  STA     EXDLPC          ;ELSE - CLEAR THE COUNTER
1646  4435  C3  46  44        10.0                  JMP     XDLPC           ;AND CONTINUE TESTING
1647
1648  4438  3A  94  46       13.0  XDLP1:  LDA     TRANSCNT        ;GET THE LOW BYTE TRANSITION COUNT
1649  443B  47                4.0                  MOV     B,A             ;TEMP SAVE
1650  443C  3A  91  46       13.0                  LDA     EXDLPC          ;GET THE TRANS. COUNT AT ERROR
1651  443F  B8                4.0                  CMP     B               ;SAME?
```

```
1652  4440  CA 05 43      10.0          JZ      T1LOOP          ;YUP - LOOP NOW!
1653  4443  C3 46 44      10.0          JMP     XDLPC           ;NO - CONTINUE TILL A MATCH...
1654
1655  4446  DB 48         10.0  XDLPC:  IN      PSTAT           ;GET THE PORT PARITY BIT
1656  4448  0F             4.0          RRC                     ;RIGHT JUSTIFY
1657  4449  0F             4.0          RRC                     ;
1658  444A  E6 01          7.0          ANI     $01             ;REMOVE OTHER BITS
1659  444C  57             4.0          MOV     D,A             ;STORE ACTUAL PARITY IN REG D
1660
1661  444D                               ROUT    ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  444D  D3 94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  444F  7F             4.0                   MOV     A,A             ;RETRY LINK
1662  4450  2A 92 46      16.0          LHLD    POINTER
1663  4453  01 EC 45      10.0          LXI     B,GXPDAT        ;LOAD BEGINNING ADDRESS OF EXPECTED DATA
1664  4456  78             4.0          MOV     A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1665  4457  2F             4.0          CMA                     ;PAIR B & C
1666  4458  47             4.0          MOV     B,A
1667  4459  79             4.0          MOV     A,C
1668  445A  2F             4.0          CMA
1669  445B  4F             4.0          MOV     C,A
1670  445C  03             6.0          INX     B
1671  445D  09            10.0          DAD     B               ;ADD POINTER AND BEGINNING OF DATA TABLE
1672  445E  D2 85 44      10.0          JNC     PPREAM          ;CHECK PREAMBLE PARITY TRACK
1673
1674  4461  2A 92 46      16.0          LHLD    POINTER         ;RESET H & L REGS
1675  4464  01 14 46      10.0          LXI     B,GDATEN        ;GET THE ENDING ADDRESS OF EXPECTED DATA
1676  4467  78             4.0          MOV     A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1677  4468  2F             4.0          CMA                     ;PAIR B & C
1678  4469  47             4.0          MOV     B,A
1679  446A  79             4.0          MOV     A,C
1680  446B  2F             4.0          CMA
1681  446C  4F             4.0          MOV     C,A
1682  446D  03             6.0          INX     B
1683  446E  09            10.0          DAD     B               ;ADD POINTER AND END OF DATA TABLE
1684  446F  DA 85 44      10.0          JC      PPOST           ;CHECK POSTAMBLE PARITY TRACK
1685
1686  4472  01 64 46      10.0  DPTST:  LXI     B,PXPTBL        ;POINT TO EXPECTED PARITY DATA TABLE
1687  4475  2A 8F 46      16.0          LHLD    DPCNT           ;POINT TO THE TABLE OFFSET NUMBER
1688  4478  23             6.0          INX     H               ;UPDATE IT
1689  4479  22 8F 46      16.0          SHLD    DPCNT           ;SAVE NEW COUNT
1690  447C  09            10.0          DAD     B               ;MAKE REAL POINTER
1691  447D  7E             7.0          MOV     A,M             ;GET EXPECTED PARITY STATE
1692  447E  A7             4.0          ANA     A               ;SET CONDITION BITS
1693  447F  FA C8 44      10.0          JM      PTYC            ;JMP IF NO COMPARE ON THIS DATA
1694  4482  C3 8A 44      10.0          JMP     PTY             ;TEST THE PARITY BIT
1695
1696                                ;HERE TO CHECK THE PREAMBLE/POSTAMBLE PARITY TRACK BIT
1697
1698  4485                          PPOST:
1699  4485  3A 8E 46      13.0  PPREAM: LDA     XDATA           ;GET THE DATA EXPECTED
1700  4488  E6 01          7.0          ANI     1               ;PARITY TRACK IS SAME
1701
1702  448A  BA             4.0  PTY:    CMP     D               ;SEE IF EXPECTED=ACTUAL
1703  448B  CA A7 44      10.0          JZ      PTSTOK          ;JMP IF OK
```

```
1704  448E                                        ROUT    EDATA           ;SAVE THE EXPECTED DATA
 (1)  448E  D3  95          10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4490  7F               4.0                         MOV     A,A             ;RETRY LINK
1705  4491  7A               4.0                 MOV     A,D             ;ERROR - GET THE ACTUAL DATA
1706  4492                                        ROUT    ADATA           ;SAVE FOR PRINTOUT
 (1)  4492  D3  94          10.0                         OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4494  7F               4.0                         MOV     A,A             ;RETRY LINK
1707
1708  4495  3A  91  46       13.0  PTY1:   LDA     EXDLPC          ;GET THE COUNT AT THE FRROR
1709  4498  A7               4.0                 ANA     A
1710  4499  C2  A2  44       10.0                 JNZ     PTY2            ;JUMP IF AT AN ERROR NOW
1711  449C  3A  94  46       13.0                 LDA     TRANSCNT        ;GET THE COUNT AT ERROR
1712  449F  32  91  46       13.0                 STA     EXDLPC          ;SAVE IT
1713  44A2                          PTY2:   ERRB    XPLOOP,PTSTOK,@4
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  44A2  CD  12  28     18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A5  04                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A6  04                                       .BYTE   @4              ;PRINT ROUTINE NUMBER
 (1)  44A7  CD  15  28     18.0                 PTSTOK::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44AA  DA  B0  44     10.0                         JC      XPLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1714                            ;>PARITY STATE RECEIVED FROM THE XMC INCORRECT
1715  44AD  C3  C8  44       10.0                 JMP     PTYC            ;JMP OVER CODE
1716
1717  44B0  3A  94  4F       13.0  XPLOOP: LDA     LPFLG           ;GET THE LOOP ON ERROR FLAG
1718  44B3  A7               4.0                 ANA     A
1719  44B4  C2  BD  44       10.0                 JNZ     XPLP1           ;JUMP IF FLAG IS SET
1720  44B7  32  91  46       13.0                 STA     EXDLPC          ;ELSE - CLEAR THE COUNTER
1721  44BA  C3  C8  44       10.0                 JMP     PTYC            ;AND CONTINUE TESTING
1722
1723  44BD  3A  94  46       13.0  XPLP1:  LDA     TRANSCNT        ;GET THE LOW BYTE TRANSITION COUNT
1724  44C0  47               4.0                 MOV     B,A             ;TEMP SAVE
1725  44C1  3A  91  46       13.0                 LDA     EXDLPC          ;GET THE TRANS. COUNT AT ERROR
1726  44C4  B8               4.0                 CMP     B               ;SAME?
1727  44C5  CA  05  43       10.0                 JZ      T1LOOP          ;YUP - LOOP NOW!
1728
1729  44C8  2A  92  46       16.0  PTYC:   LHLD    POINTER         ;NO-UPDATE THE
1730  44CB  23               6.0                 INX     H               ;TABLE POINTER
1731  44CC  22  92  46       16.0                 SHLD    POINTER         ;ADDRESS
1732
1733  44CF  06  0E           7.0                 MVI     B,14            ;LOAD THE CLOCK COUNT
1734  44D1  CD  60  45       18.0  1$:     CALL    CLKSYS          ;CLOCK THE WRITE PATH
1735  44D4  05               4.0                 DCR     B               ;DECREMENT THE COUNT
1736  44D5  C2  D1  44       10.0                 JNZ     1$              ;CONTINUE UNTIL 8 CLOCKS
1737  44D8  DB  D0           10.0                 IN      WMCSTA
1738  44DA  E6  01           7.0                  ANI     X.WCLK
1739  44DC  C2  E4  44       10.0                 JNZ     PRECN3
```

```
1741  44DF                                    ERR     T1LOOP,PRECN3,@4
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44DF   CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E2   05                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E3   04                                        .BYTE   @4
 (1)  44E4   CD  15  28        18.0            PRECN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44E7   DA  05  43        10.0                    JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1742                                    ;>TRANSLATOR 'XL WCLK'' NOT SET AFTER EIGHT CLOCK CYCLES
1743                                    ;>IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1744
1745  44EA   2A  92  46        16.0            LHLD    POINTER         ;GET THE TABLE POINTER
1746  44ED   01  63  46        10.0            LXI     B,GTBLEN        ;GET END OF TABLE ADDRESS
1747  44F0   78                  4.0            MOV     A,B             ;FROM THE
1748  44F1   2F                  4.0            CMA                     ;TWO'S COMPLIMENT
1749  44F2   47                  4.0            MOV     B,A             ;OF THE TABLE END
1750  44F3   79                  4.0            MOV     A,C             ;ADDRESS
1751  44F4   2F                  4.0            CMA
1752  44F5   4F                  4.0            MOV     C,A
1753  44F6   03                  6.0            INX     B
1754  44F7   09                 10.0            DAD     B               ;END OF TABLE?
1755  44F8   DA  FE  44          10.0            JC      RECEND          ;YES
1756  44FB   C3  E7  43          10.0            JMP     BGNPRE
1757
1758  44FE   06  FF               7.0  RECEND:  MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1759  4500   CD  60  45          18.0  1$:      CALL    CLKSYS          ;CLOCK THE SYSTEM
1760  4503   DB  D0              10.0            IN      WMCSTA          ;INPUT THE WMC STATUS
1761  4505   E6  40               7.0            ANI     W.DONN          ;WMC DONE?
1762  4507   CA  1A  45          10.0            JZ      RECCN1          ;YES-EXIT
1763  450A   DB  D0              10.0            IN      WMCSTA          ;GET WMC STATUS
1764  450C   E6  80               7.0            ANI     X.DONN          ;XMC DONE
1765  450E   CA  3E  45          10.0            JZ      SEQERR          ;YES-ERROR
1766  4511   05                  4.0            DCR     B               ;NO-DECREMENT THE COUNT
1767  4512   C2  00  45          10.0            JNZ     1$              ;CONTINUE IF NOT ZERO
1768  4515                                    ERR     RECEND,RECCN1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4515   CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4518   06                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4519   00                                        .BYTE
 (1)  451A   CD  15  28        18.0            RECCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  451D   DA  FE  44        10.0                    JC      RECEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1769                                    ;>'WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1770
1771  4520   06  FF               7.0            MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1772  4522   CD  60  45          18.0  3$:      CALL    CLKSYS          ;CLOCK THE SYSTEM
1773  4525   DB  D0              10.0            IN      WMCSTA          ;INPUT THE WMC STATUS
1774  4527   E6  80               7.0            ANI     X.DONN          ;XMC-DONE?
1775  4529   CA  35  45          10.0            JZ      RECCN3          ;YES-EXIT
1776  452C   05                  4.0            DCR     B               ;NO-DECREMENT THE COUNT
1777  452D   C2  22  45          10.0            JNZ     3$              ;CONTINUE IF NOT ZERO
```

```
1779  4530                                          ERR     T1LOOP,RECCN3
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4530  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4533  07                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4534  00                                             .BYTE
 (1)  4535  CD  15  28        18.0                  RECCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4538  DA  05  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1780                                        ;>''XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1781
1782  453B  C3  49  45        10.0          JMP     XMEND
1783
1784  453E                                  SEQERR: ERR     T1LOOP,RECCN2
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453E  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4541  08                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4542  00                                             .BYTE
 (1)  4543  CD  15  28        18.0                  RECCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4546  DA  05  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1785                                        ;>''XMC NOT DONE'' RESET BEFORE ''XMC NOT DONE'' RESET
1786
1787  4549                                  XMEND:  ENDTST  T1LOOP
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4549                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4549  CD  06  28        18.0                          CALL    REQST
 (2)  454C  00                                             .BYTE                   ;DATA PATTERN NUMBER
 (2)  454D  00  00                                         .WORD                   ;SYSTEM ''*'' COUNT
 (2)  454F  00  00                                         .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4551  00                                             .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4552  07                                             .BYTE   7               ;REQUEST CODE
 (1)  4553  3A  9A  4F        13.0                  LDA     ITERA           ;GET ITERATION COUNT
 (1)  4556  3D                 4.0                  DCR     A               ;DOWNCOUNT
 (1)  4557  32  9A  4F        13.0                  STA     ITERA           ;SAVE COUNT
 (1)  455A  F2  05  43        10.0                  JP      T1LOOP          ;DO TEST UNTIL TILL = 0
1788
1789  455D  C3  18  28        10.0  EXIT:   JMP     TSTEND
1790
```

```
                                        .SBTTL  SUBROUTINE CLKSYS

                                        SD
                                        ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
 (1)                                    ;*THIS SUBROUTINE WILL GENERATE ONE SINGLE STEP USING THE CONTENTS OF
                                        ;*CCTLWD AS THE CLOCK CONTROL BYTE.
                                        S
 (1)                                    ; ************************************************************************

4560  3A  96  46        13.0  CLKSYS: LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
4563  F6  40             7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
4565  D3  F0            10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
4567  E6  BF             7.0          ANI     @277            ;STRIP OFF CLOCK BIT
4569  D3  F0            10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
456B  C9                10.0          RET                     ;EXIT-CLOCK CYCLE
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************
                              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                              12.0          PUSH    B
                               7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
                              10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
                               7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
                              10.0          OUT     MBSEL           ;RESET TO THIS PORT
                               7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                              10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                               4.0          XRA     A               ;CLEAR TU COMMAND A
                              10.0          OUT     TCMD            ;
                               7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                              10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                               7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
                              10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
                               4.0          XRA     A
                              10.0          OUT     TAMT            ;CLEAR AMTIE WORD
                              10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                              10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
                               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                               4.0          MOV     A,B
                               7.0          CPI     4               ;DONE?
                              10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                              10.0          POP     B               ;RESET B & C
                              10.0          POP     PSW             ;ALL DONE
                              10.0          RET                     ;EXIT
```

```
1850  4597  FF                    GXPTBL: .BYTE   $FF             ;PREAMBLE TERM GROUP (10101)
1851                                                              ;CHARACTER 1 = ALL ONES
1852  4598  FF                            .BYTE   $FF             ;PREAMBLE TERM GROUP
1853                                                              ;CHARACTER 2 = ALL ZEROS
1854  4599  00                            .BYTE   $00             ;PREAMBLE TERM GROUP
1855                                                              ;CHARACTER 3 = ALL ONES
1856  459A  00                            .BYTE   $00             ;PREAMBLE TERM GROUP
1857                                                              ;CHARACTER 4 = ALL ZEROS
1858  459B  FF                            .BYTE   $FF             ;PREAMBLE TERM GROUP
1859                                                              ;CHARACTER 5 = ALL ONES
1860
1861  459C  FF                            .BYTE   $FF             ;PREAMBLE SECOND GROUP (01111)
1862                                                              ;CHARACTER 1 = ALL ZEROS
1863  459D  00                            .BYTE   $00             ;PREAMBLE SECOND GROUP
1864                                                              ;CHARACTER 2 = ALL ONES
1865  459E  FF                            .BYTE   $FF             ;PREAMBLE SECOND GROUP
1866                                                              ;CHARACTER 3 = ALL ONES
1867  459F  00                            .BYTE   $00             ;PREAMBLE SECOND GROUP
1868                                                              ;CHARACTER 4 = ALL ONES
1869  45A0  FF                            .BYTE   $FF             ;PREAMBLE SECOND GROUP
1870                                                              ;CHARACTER 5 = ALL ONES
1871                                                              ;
1872  45A1  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 1 (14 GRPS OF 11111)
1873                                                              ;CHARACTER 1 = ALL ONES
1874  45A2  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 1
1875                                                              ;CHARACTER 2 = ALL ONES
1876  45A3  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 1
1877                                                              ;CHARACTER 3 = ALL ONES
1878  45A4  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 1
1879                                                              ;CHARACTER 4 - ALL ONES
1880  45A5  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 1
1881                                                              ;CHARACTER 5 = ALL ONES
1882                                                              ;
1883  45A6  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 2
1884                                                              ;CHARACTER 1 = ALL ONES
1885  45A7  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 2
1886                                                              ;CHARACTER 2 = ALL ONES
1887  45A8  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 2
1888                                                              ;CHARACTER 3 = ALL ONES
1889  45A9  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 2
1890                                                              ;CHARACTER 4 = ALL ONES
1891  45AA  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 2
1892                                                              ;CHARACTER 5 = ALL ONES
1893                                                              ;
1894  45AB  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 3
1895                                                              ;CHARACTER 1 = ALL ONES
1896  45AC  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 3
1897                                                              ;CHARACTER 2 = ALL ONES
1898  45AD  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 3
1899                                                              ;CHARACTER 3 = ALL ONES
1900  45AE  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 3
1901                                                              ;CHARACTER 4 - ALL ONES
1902  45AF  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 3
1903                                                              ;CHARACTER 5 = ALL ONES
```

```
1904                                                    ;
1905  45B0  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 4
1906                                                    ;CHARACTER 1 = ALL ONES
1907  45B1  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 4
1908                                                    ;CHARACTER 2 = ALL ONES
1909  45B2  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 4
1910                                                    ;CHARACTER 3 = ALL ONES
1911  45B3  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 4
1912                                                    ;CHARACTER 4 = ALL ONES
1913  45B4  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 4
1914                                                    ;CHARACTER 5 = ALL ONES
1915                                                    ;
1916  45B5  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 5
1917                                                    ;CHARACTER 1 = ALL ONES
1918  45B6  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 5
1919                                                    ;CHARACTER 2 = ALL ONES
1920  45B7  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 5
1921                                                    ;CHARACTER 3 = ALL ONES
1922  45B8  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 5
1923                                                    ;CHARACTER 4 - ALL ONES
1924  45B9  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 5
1925                                                    ;CHARACTER 5 = ALL ONES
1926                                                    ;
1927  45BA  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 6
1928                                                    ;CHARACTER 1 = ALL ONES
1929  45BB  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 6
1930                                                    ;CHARACTER 2 = ALL ONES
1931  45BC  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 6
1932                                                    ;CHARACTER 3 = ALL ONES
1933  45BD  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 6
1934                                                    ;CHARACTER 4 = ALL ONES
1935  45BE  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 6
1936                                                    ;CHARACTER 5 = ALL ONES
1937                                                    ;
1938  45BF  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 7
1939                                                    ;CHARACTER 1 = ALL ONES
1940  45C0  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 7
1941                                                    ;CHARACTER 2 = ALL ONES
1942  45C1  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 7
1943                                                    ;CHARACTER 3 = ALL ONES
1944  45C2  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 7
1945                                                    ;CHARACTER 4 - ALL ONES
1946  45C3  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 7
1947                                                    ;CHARACTER 5 = ALL ONES
1948                                                    ;
1949  45C4  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 8
1950                                                    ;CHARACTER 1 = ALL ONES
1951  45C5  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 8
1952                                                    ;CHARACTER 2 = ALL ONES
1953  45C6  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 8
1954                                                    ;CHARACTER 3 = ALL ONES
1955  45C7  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 8
1956                                                    ;CHARACTER 4 = ALL ONES
1957  45C8  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 8
```

```
1958                                                    ;CHARACTER 5 = ALL ONES
1959                                                    ;
1960  45C9  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 9
1961                                                    ;CHARACTER 1 = ALL ONES
1962  45CA  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 9
1963                                                    ;CHARACTER 2 = ALL ONES
1964  45CB  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 9
1965                                                    ;CHARACTER 3 = ALL ONES
1966  45CC  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 9
1967                                                    ;CHARACTER 4 = ALL ONES
1968  45CD  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 9
1969                                                    ;CHARACTER 5 = ALL ONES
1970                                                    ;
1971  45CE  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 10
1972                                                    ;CHARACTER 1 = ALL ONES
1973  45CF  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 10
1974                                                    ;CHARACTER 2 = ALL ONES
1975  45D0  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 10
1976                                                    ;CHARACTER 3 = ALL ONES
1977  45D1  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 10
1978                                                    ;CHARACTER 4 = ALL ONES
1979  45D2  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 10
1980                                                    ;CHARACTER 5 = ALL ONES
1981                                                    ;
1982  45D3  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 11
1983                                                    ;CHARACTER 1 = ALL ONES
1984  45D4  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 11
1985                                                    ;CHARACTER 2 = ALL ONES
1986  45D5  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 11
1987                                                    ;CHARACTER 3 = ALL ONES
1988  45D6  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 11
1989                                                    ;CHARACTER 4 = ALL ONES
1990  45D7  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 11
1991                                                    ;CHARACTER 5 = ALL ONES
1992                                                    ;
1993  45D8  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 12
1994                                                    ;CHARACTER 1 = ALL ONES
1995  45D9  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 12
1996                                                    ;CHARACTER 2 = ALL ONES
1997  45DA  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 12
1998                                                    ;CHARACTER 3 = ALL ONES
1999  45DB  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 12
2000                                                    ;CHARACTER 4 = ALL ONES
2001  45DC  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 12
2002                                                    ;CHARACTER 5 = ALL ONES
2003                                                    ;
2004  45DD  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 13
2005                                                    ;CHARACTER 1 = ALL ONES
2006  45DE  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 13
2007                                                    ;CHARACTER 2 = ALL ONES
2008  45DF  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 13
2009                                                    ;CHARACTER 3 = ALL ONES
2010  45E0  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 13
2011                                                    ;CHARACTER 4 = ALL ONES
```

```
2012  45E1  00                        .BYTE   $00            ;PREAMBLE SYNC GROUP 13
2013                                                         ;CHARACTER 5 = ALL ONES
2014                                                         ;
2015  45E2  FF                        .BYTE   $FF            ;PREAMBLE SYNC GROUP 14
2016                                                         ;CHARACTER 1 = ALL ONES
2017  45E3  00                        .BYTE   $00            ;PREAMBLE SYNC GROUP 14
2018                                                         ;CHARACTER 2 = ALL ONES
2019  45E4  FF                        .BYTE   $FF            ;PREAMBLE SYNC GROUP 14
2020                                                         ;CHARACTER 3 = ALL ONES
2021  45E5  00                        .BYTE   $00            ;PREAMBLE SYNC GROUP 14
2022                                                         ;CHARACTER 4 = ALL ONES
2023  45E6  FF                        .BYTE   $FF            ;PREAMBLE SYNC GROUP 14
2024                                                         ;CHARACTER 5 = ALL ONES
2025                                                         ;
2026  45E7  FF                        .BYTE   $FF            ;MARK ONE GROUP
2027                                                         ;CHARACTER 1 = ALL ZEROS
2028  45E8  FF                        .BYTE   $FF            ;MARK ONE GROUP
2029                                                         ;CHARACTER 2 = ALL ZEROS
2030  45E9  00                        .BYTE   $00            ;MARK ONE GROUP
2031                                                         ;CHARACTER 3 = ALL ONES
2032  45EA  FF                        .BYTE   $FF            ;MARK ONE GROUP
2033                                                         ;CHARACTER 4 = ALL ONES
2034  45EB  00                        .BYTE   $00            ;MARK ONE GROUP
2035                                                         ;CHARACTER 5 = ALL ONES
2036                                                         ;
2037                          ;HERE IS THE DATA GROUP - 7 BYTES OF ALL 0'S (BYTE #86.)
2038                          ;ECC FOR DATA GRP = 0.
2039
2040  45EC  FF                GXPDAT: .BYTE   $FF            ;1ST CHAR IN 1ST HALF OF DATA GROUP - P = 0
2041  45ED  00                        .BYTE   0              ;2ND CHAR - P = 1
2042  45EE  00                        .BYTE   0              ;3RD CHAR - P = 1
2043  45EF  00                        .BYTE   0              ;4TH CHAR - P = 1
2044  45F0  FF                        .BYTE   $FF            ;5TH CHAR - P = 1
2045
2046  45F1  00                        .BYTE   0              ;1ST CHAR IN 2ND HALF OF DATA GROUP - P = 0
2047  45F2  FF                        .BYTE   $FF            ;2ND CHAR - P = 1
2048  45F3  FF                        .BYTE   $FF            ;3RD CHAR - P = 1
2049  45F4  FF                        .BYTE   $FF            ;4TH CHAR - P = 1
2050  45F5  00                        .BYTE   0              ;5TH CHAR (ECC CHAR =0) - P = 1
2051
2052                          ;END MARK = 11111 (ALL 1'S)
2053
2054  45F6  FF                        .BYTE   $FF            ;END MARK GROUP
2055                                                         ;CHARACTER 1 = ALL ONES
2056  45F7  00                        .BYTE   $00            ;END MARK GROUP
2057                                                         ;CHARACTER 2 = ALL ONES
2058  45F8  FF                        .BYTE   $FF            ;END MARK GROPU
2059                                                         ;CHARACTER 3 = ALL ONES
2060  45F9  00                        .BYTE   $00            ;END MARK GROUP
2061                                                         ;CHARACTER 4 = ALL ONES
2062  45FA  FF                        .BYTE   $FF            ;END MARK GROUP
2063                                                         ;CHARACTER 5 = ALL ONES
2064                                                         ;
2065
```

```
2066
2067                                    ;4 PAD BYTES
2068
2069  45FB  00                                .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION - 1ST HALF
                                                                      ;CHARACTER 1
2070
2071  45FC  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 2
2072
2073  45FD  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 3
2074
2075  45FE  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 4
2076
2077  45FF  00                                .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 5
2078
2079                                    ;2 PAD BYTES
2080                                    ;ACRC = 323
2081                                    ;ECC = 225
2082  4600  FF                                .BYTE   @377            ;RESIDUAL GROUP - AFTER TRANSLATION - 2ND HALF
                                                                      ;CHARACTER 1
2083
2084  4601  D3                                .BYTE   @323            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 2
2085
2086  4602  D3                                .BYTE   @323            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 3
2087
2088  4603  04                                .BYTE   @004            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 4
2089
2090  4604  B9                                .BYTE   @271            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 5
2091
2092                                    ;4 CRC BYTES OF 136
2093  4605  18                                .BYTE   @030            ;CRC GROUP - AFTER TRANSLATION - 1ST HALF
                                                                      ;CHARACTER 1
2094
2095  4606  E7                                .BYTE   @347            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 2
2096
2097  4607  B9                                .BYTE   @271            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 3
2098
2099  4608  E7                                .BYTE   @347            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 4
2100
2101  4609  18                                .BYTE   @030            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 5
2102
2103                                    ;2 CRC OF 136
2104                                    ;RESID OF 006
2105                                    ;ECC OF 352
2106  460A  A9                                .BYTE   @251            ;CRC GROUP - AFTER TRANSLATION - 2ND HALF
                                                                      ;CHARACTER 6
2107
2108  460B  56                                .BYTE   @126            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 7
2109
2110  460C  08                                .BYTE   @010            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 8
2111
2112  460D  BE                                .BYTE   @276            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 9
2113
2114  460E  55                                .BYTE   @125            ;CRC GROUP - AFTER TRANSLATION
                                                                      ;CHARACTER 10
2115                                                                  ;
2116
2117
2118
2119  460F  AA                                .BYTE   @252            ;MARK TWO
```

```
2120
2121  4610  55                          .BYTE   @125    ;CHARACTER 1 = ALL ONES
2122                                                    ;MARK TWO
2123  4611  AA                          .BYTE   @252    ;CHARACTER 2 = ALL ONES
2124                                                    ;MARK TWO
2125  4612  AA                          .BYTE   @252    ;CHARACTER 3 = ALL ONES
2126                                                    ;MARK TWO
2127  4613  AA                          .BYTE   @252    ;CHARACTER 4 = ALL ZEROS
2128                                                    ;MARK TWO
2129                                                    ;CHARACTER 5 = ALL ZEROS
2130  4614  55                GDATEN:   .BYTE   @125    ;
2131                                                    ;POSTAMBLE SYNC GROUP 1
2132  4615  AA                          .BYTE   @252    ;CHARACTER 1 = ALL ONES
2133                                                    ;POSTAMBLE SYNC GROUP 1
2134  4616  55                          .BYTE   @125    ;CHARACTER 2 - ALL ONES
2135                                                    ;POSTAMBLE SYNC GROUP 1
2136  4617  AA                          .BYTE   @252    ;CHARACTER 3 = ALL ONES
2137                                                    ;POSTAMBLE SYNC GROUP 1
2138  4618  55                          .BYTE   @125    ;CHARACTER 4 = ALL ONES
2139                                                    ;POSTAMBLE SYNC GROUP 1
2140                                                    ;CHARACTER 5 = ALL ONES
2141  4619  AA                          .BYTE   @252    ;
2142                                                    ;POSTAMBLE SYNC GROUP 2
2143  461A  55                          .BYTE   @125    ;CHARACTER 1 = ALL ONES
2144                                                    ;POSTAMBLE SYNC GROUP 2
2145  461B  AA                          .BYTE   @252    ;CHARACTER 2 - ALL ONES
2146                                                    ;POSTAMBLE SYNC GROUP 2
2147  461C  55                          .BYTE   @125    ;CHARACTER 3 = ALL ONES
2148                                                    ;POSTAMBLE SYNC GROUP 2
2149  461D  AA                          .BYTE   @252    ;CHARACTER 4 = ALL ONES
2150                                                    ;POSTAMBLE SYNC GROUP 2
2151                                                    ;CHARACTER 5 = ALL ONES
2152  461E  55                          .BYTE   @125    ;
2153                                                    ;POSTAMBLE SYNC GROUP 3
2154  461F  AA                          .BYTE   @252    ;CHARACTER 1 = ALL ONES
2155                                                    ;POSTAMBLE SYNC GROUP 3
2156  4620  55                          .BYTE   @125    ;CHARACTER 2 - ALL ONES
2157                                                    ;POSTAMBLE SYNC GROUP 3
2158  4621  AA                          .BYTE   @252    ;CHARACTER 3 = ALL ONES
2159                                                    ;POSTAMBLE SYNC GROUP 3
2160  4622  55                          .BYTE   @125    ;CHARACTER 4 = ALL ONES
2161                                                    ;POSTAMBLE SYNC GROUP 3
2162                                                    ;CHARACTER 5 = ALL ONES
2163  4623  AA                          .BYTE   @252    ;
2164                                                    ;POSTAMBLE SYNC GROUP 4
2165  4624  55                          .BYTE   @125    ;CHARACTER 1 = ALL ONES
2166                                                    ;POSTAMBLE SYNC GROUP 4
2167  4625  AA                          .BYTE   @252    ;CHARACTER 2 - ALL ONES
2168                                                    ;POSTAMBLE SYNC GROUP 4
2169  4626  55                          .BYTE   @125    ;CHARACTER 3 = ALL ONES
2170                                                    ;POSTAMBLE SYNC GROUP 4
2171  4627  AA                          .BYTE   @252    ;CHARACTER 4 = ALL ONES
2172                                                    ;POSTAMBLE SYNC GROUP 4
2173                                                    ;CHARACTER 5 = ALL ONES
                                                        ;
```

```
2174  4628  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 5
2175                                        ;CHARACTER 1 = ALL ONES
2176  4629  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 5
2177                                        ;CHARACTER 2 - ALL ONES
2178  462A  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 5
2179                                        ;CHARACTER 3 = ALL ONES
2180  462B  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 5
2181                                        ;CHARACTER 4 = ALL ONES
2182  462C  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 5
2183                                        ;CHARACTER 5 = ALL ONES
2184                                        ;
2185  462D  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 6
2186                                        ;CHARACTER 1 = ALL ONES
2187  462E  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 6
2188                                        ;CHARACTER 2 - ALL ONES
2189  462F  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 6
2190                                        ;CHARACTER 3 = ALL ONES
2191  4630  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 6
2192                                        ;CHARACTER 4 = ALL ONES
2193  4631  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 6
2194                                        ;CHARACTER 5 = ALL ONES
2195                                        ;
2196  4632  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 7
2197                                        ;CHARACTER 1 = ALL ONES
2198  4633  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 7
2199                                        ;CHARACTER 2 - ALL ONES
2200  4634  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 7
2201                                        ;CHARACTER 3 = ALL ONES
2202  4635  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 7
2203                                        ;CHARACTER 4 = ALL ONES
2204  4636  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 7
2205                                        ;CHARACTER 5 = ALL ONES
2206                                        ;
2207  4637  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 8
2208                                        ;CHARACTER 1 = ALL ONES
2209  4638  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 8
2210                                        ;CHARACTER 2 - ALL ONES
2211  4639  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 8
2212                                        ;CHARACTER 3 = ALL ONES
2213  463A  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 8
2214                                        ;CHARACTER 4 = ALL ONES
2215  463B  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 8
2216                                        ;CHARACTER 5 = ALL ONES
2217                                        ;
2218  463C  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 9
2219                                        ;CHARACTER 1 = ALL ONES
2220  463D  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 9
2221                                        ;CHARACTER 2 - ALL ONES
2222  463E  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 9
2223                                        ;CHARACTER 3 = ALL ONES
2224  463F  AA          .BYTE   @252        ;POSTAMBLE SYNC GROUP 9
2225                                        ;CHARACTER 4 = ALL ONES
2226  4640  55          .BYTE   @125        ;POSTAMBLE SYNC GROUP 9
2227                                        ;CHARACTER 5 = ALL ONES
```

```
2228
2229  4641  AA          .BYTE   @252    ;
                                        ;POSTAMBLE SYNC GROUP 10
                                        ;CHARACTER 1 = ALL ONES
2230
2231  4642  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 10
                                        ;CHARACTER 2 - ALL ONES
2232
2233  4643  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 10
                                        ;CHARACTER 3 = ALL ONES
2234
2235  4644  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 10
                                        ;CHARACTER 4 = ALL ONES
2236
2237  4645  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 10
                                        ;CHARACTER 5 = ALL ONES
2238
2239                                    ;
2240  4646  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 11
                                        ;CHARACTER 1 = ALL ONES
2241
2242  4647  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 11
                                        ;CHARACTER 2 - ALL ONES
2243
2244  4648  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 11
                                        ;CHARACTER 3 = ALL ONES
2245
2246  4649  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 11
                                        ;CHARACTER 4 = ALL ONES
2247
2248  464A  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 11
                                        ;CHARACTER 5 = ALL ONES
2249
2250                                    ;
2251  464B  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 12
                                        ;CHARACTER 1 = ALL ONES
2252
2253  464C  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 12
                                        ;CHARACTER 2 - ALL ONES
2254
2255  464D  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 12
                                        ;CHARACTER 3 = ALL ONES
2256
2257  464E  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 12
                                        ;CHARACTER 4 = ALL ONES
2258
2259  464F  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 12
                                        ;CHARACTER 5 = ALL ONES
2260
2261                                    ;
2262  4650  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 13
                                        ;CHARACTER 1 = ALL ONES
2263
2264  4651  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 13
                                        ;CHARACTER 2 - ALL ONES
2265
2266  4652  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 13
                                        ;CHARACTER 3 = ALL ONES
2267
2268  4653  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 13
                                        ;CHARACTER 4 = ALL ONES
2269
2270  4654  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 13
                                        ;CHARACTER 5 = ALL ONES
2271
2272                                    ;
2273  4655  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 14
                                        ;CHARACTER 1 = ALL ONES
2274
2275  4656  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 14
                                        ;CHARACTER 2 - ALL ONES
2276
2277  4657  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 14
                                        ;CHARACTER 3 = ALL ONES
2278
2279  4658  55          .BYTE   @125    ;POSTAMBLE SYNC GROUP 14
                                        ;CHARACTER 4 = ALL ONES
2280
2281  4659  AA          .BYTE   @252    ;POSTAMBLE SYNC GROUP 14
```

```
2282                                                        ;CHARACTER 5 = ALL ONES
2283                                                        ;
2284  465A  55                          .BYTE   @125       ;POSTAMBLE SECOND GROUP
2285                                                        ;CHARACTER 1 = ALL ONES
2286  465B  AA                          .BYTE   @252       ;POSTAMBLE SECOND GROUP
2287                                                        ;CHARACTER 2 = ALL ONES
2288  465C  55                          .BYTE   @125       ;POSTAMBLE SECOND GROUP
2289                                                        ;CHARACTER 3 = ALL ONES
2290  465D  AA                          .BYTE   @252       ;POSTAMBLE SECOND GROUP
2291                                                        ;CHARACTER 4 = ALL ONES
2292  465E  AA                          .BYTE   @252       ;POSTAMBLE SECOND GROUP
2293                                                        ;CHARACTER 5 = ALL ZEROS
2294                                                        ;
2295  465F  55                          .BYTE   @125       ;POSTAMBLE TERM GROUP
2296                                                        ;CHARACTER 1 = ALL ONES
2297  4660  55                          .BYTE   @125       ;POSTAMBLE TERM GROUP
2298                                                        ;CHARACTER 2 = ALL ZEROS
2299  4661  AA                          .BYTE   @252       ;POSTAMBLE TERM GROUP
2300                                                        ;CHARACTER 3 = ALL ONES
2301  4662  AA                          .BYTE   @252       ;POSTAMBLE TERM GROUP
2302                                                        ;CHARACTER 4 = ALL ZEROS
2303  4663  AA                GTBLEN:   .BYTE   @252       ;POSTAMBLE TERM GROUP
2304                                                        ;CHARACTER 5 = 0'S TO RESET TRACKS
```

```
                                        .SBTTL  TABLE EXPECTED PARITY

                                        ;HERE IS THE TABLE OF EXPECTED PARITY STATES FOR DATA & RESIDUAL GROUPS
                                        ;ALL BYTES IN DATA GRP=0 (EVEN ECC) SO 'P' TRACK 4 X 5 IS 0001 = 01111

4664  00                                PXPTBL: .BYTE   0               ;1ST ENTRY IS A DUMMY
4665  00                                        .BYTE   0               ;'P' TRANSLATION ACTS SAME AS OTHER
4666  01                                        .BYTE   1               ;TRACKS
4667  00                                        .BYTE   0
4668  01                                        .BYTE   1
4669  00                                        .BYTE   0
466A  00                                        .BYTE   0
466B  01                                        .BYTE   1
466C  00                                        .BYTE   0
466D  01                                        .BYTE   1
466E  00                                        .BYTE   0

                                        ;SKIP OVER END MARK GRP

466F  FF                                        .BYTE   @377
4670  FF                                        .BYTE   @377
4671  FF                                        .BYTE   @377
4672  FF                                        .BYTE   @377
4673  FF                                        .BYTE   @377

                                        ;RESIDUAL GRP
                                        ;1ST GRP IS 4 PADS SO 4 X 5 IS 0000 = 11001

4674  01                                        .BYTE   1
4675  00                                        .BYTE   0
4676  01                                        .BYTE   1
4677  00                                        .BYTE   0
4678  01                                        .BYTE   1

                                        ;2ND GRP IS 2 PAD BYTES + ACRC OF 323 + ECC OF 225
                                        ;'P' 4 X 5 IS 1101 = 01101

4679  01                                        .BYTE   1
467A  00                                        .BYTE   0
467B  01                                        .BYTE   1
467C  01                                        .BYTE   1
467D  00                                        .BYTE   0

                                        ;CRC GRP 1ST HALF IS 4 CRC BYTES OF 136
                                        ;'P' 4 X 5 IS 0000 = 11001

467E  01                                        .BYTE   1
467F  00                                        .BYTE   0
4680  00                                        .BYTE   0
4681  00                                        .BYTE   0
4682  01                                        .BYTE   1

                                        ;2ND HALF OF CRC GRP IS 2 CRC OF 136 + RESID BYTE OF 006 + ECC BYTE OF 352
                                        ;'P' 4 X 5 IS 0010 = 10010
```

```
2361
2362  4683  00                        .BYTE   0
2363  4684  00                        .BYTE   0
2364  4685  00                        .BYTE   0
2365  4686  01                        .BYTE   1
2366  4687  01                        .BYTE   1
2367
2368                            ;HERE IS THE MARK-2 PARITY TABLE
2369
2370  4688  00                        .BYTE   0               ;MARK 2 IS 11100
2371  4689  01                        .BYTE   1
2372  468A  00                        .BYTE   0
2373  468B  00                        .BYTE   0
2374  468C  00                        .BYTE   0
2375
2376
2377
2378
2379                            .SBTTL  PROGRAM VARIABLES
2380
2381  468D  00                  UNITMP: .BYTE   0               ;UNIT MAP
2382  468E  00                  XDATA:  .BYTE   0               ;COPY OF EXPECTED DATA
2383  468F  00  00              DPCNT:  .WORD   0               ;PARITY TEST COUNTER
2384  4691  00                  EXDLPC: .BYTE   0               ;TRANSITION COUNT AT ERROR
2385  4692  00  00              POINTER: .WORD  0               ;POINTER TO EXPECTED DATA TABLE
2386  4694  00  00              TRANSCNT: .WORD 0               ;TRANSITION COUNT
2387  4696  00                  CCTLWD: .BYTE   0               ;CLOCK CONTROL WORD
2388
2389        0000                      .END
```

```
A      =%0007          ABOCNO  43DE  G       ADATA  = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098           ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000          BADST  = 0090          BGNPRE  43E7
BIT0   = 0001          BIT1   = 0002          BIT15  = 8000          BIT2   = 0004
BIT3   = 0008          BIT4   = 0010          BIT5   = 0020          BIT6   = 0040
BIT7   = 0080          BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A
BRKRAM= 4F10           BRKSTR= 4E60           BRKXCT= 4F00           BSAVE   4F9C
BYTCNT= 00D4           BYTEH   4F24           BYTEL   4F23           C      =%0001
CASCT   4F21           CASCTL= 00A0           CASSTA= 00A0           CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1           CBYTH  = 008B          CBYTL  = 008A
CCTLWD  4696           CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028           CH0TIE= 0020           CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023           CH4TIE= 0024           CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027           CKLOP  = 2815          CLEAR   456C
CLK     43CB           CLKCTL= 00F0           CLKRS   43BE          CLKSYS  4560
CLK1    43C9           CLOCK   4F26           CLRLP   4570          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7           CRCWRD= 0018           CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082          C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040
C.WCS  = 0002          D      =%0002          DATACT= 00D0           DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB           DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 0008          DONE1  = 0045          DONINT= 0010
DPCNT   468F           DPTST   4472           DSAVE   4F9E          DSE    = 0006
DUMMY   4358  G        D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          E      =%0003          ECCBAD= 0042          ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095
EOTCLR= 0003           ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90          ERRCNT= 00D6
ESAVE   4F9F           EXDLPC  4691           EXIT    455D          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001           E.TTEC= 0002          E.UNC  = 0004
FIFORD= 006A           FORMAT  4F25           FOUND   4361          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002           GDATEN  4614          GOODTM= 0092
GTBLEN  4663           GXPDAT  45EC           GXPTBL  4597          H      =%0004
HLSAVE  4FA0           IE     = 0008          INTSTA= 00E0          ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020
I7.5   = 0040          KCALL  = 005F          KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8          KEY1   = 0078
KEY10  = 006D          KEY11  = 006E          KEY12  = 006F          KEY13  = 005C
KEY14  = 005D          KEY15  = 005E          KEY16  = 005F          KEY17  = 003C
KEY18  = 003D          KEY19  = 003E          KEY2   = 0079          KEY20  = 003F
KEY3   = 007A          KEY4   = 007B          KEY5   = 0074          KEY6   = 0075
KEY7   = 0076          KEY8   = 0077          KEY9   = 006C          KINTA  = 006F
```

```
KLDAD = 003D    KN0   = 003C    KN1   = 005C    KN2   = 005D
KN3   = 005E    KN4   = 006C    KN5   = 006D    KN6   = 006E
KN7   = 0074    KN8   = 0075    KN9   = 0076    KU2   = 0079
KU3   = 007A    KU8   = 007?    L     =%0005    LBLANK= 000F
LCE   = 000B    LCH   = 000C    LCL   = 000D    LCP   = 000E
LC0   = 0000    LC1   = 0001    LC2   = 0002    LC3   = 0003
LC4   = 0004    LC5   = 0005    LC6   = 0006    LC7   = 0007
LC8   = 0008    LC9   = 0009    LDLEDA= 00CA    LDLEDB= 00CB
LDLEDC= 00CC    LDLEDD= 00CD    LDLEDE= 00CE    LDLEDF= 00CF
LKDIAG= 2800    LKKBD = 004C    LKKEY = 0049    LKLWMG= 0058
LKLWMP= 0055    LKLWPG= 0052    LKLWPP= 004F    LKMOD7= 0046
LKOPR = 0046    LPFLG   4F94    LPNUM   4F92    M     =%0006
MBSEL = 00E0    MB.A  = 0008    MB.B  = 0004    MEMTOP= 4FFF
MINUS = 000A    MM    = 8000    MSE   = 0008    MSGN  = 0008
MTACLR= 0000    MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001
MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002
MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001
MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010    MT.Z  = 0008
M.ATA = 0080    M.CAPE= 0020    M.CONT= 0080    M.DEM = 0020
M.EBL = 0010    M.EXC = 0008    M.FAIL= 0008    M.ILR = 0010
M.INIT= 0010    M.OCC = 0020    M.ONLI= 0001    M.PE  = 0040
M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008    M.RUN = 0004
M.SCLK= 0001    M.TRA = 0040    M.UNIT= 0007    M.WCLK= 0040
M.WCLN= 0080    M.WREN= 0080    M5.5  = 0001    M6.5  = 0002
M7.5  = 0004    NOTCAP= 0088    OKAY  = 00FF    OPRRAM= 4300
OPSTRT= 0058    OPVER = 0040    PADCNT= 00D5    PADCRC= 0080
PDIAG = 0048    PEID  = 008A    PENAB = 004C    PESET = 0001
PL    = 00B1    POINTE  4692    PPOST   4485    PPREAM  4485
PRDD  = 004C    PRECN1  4422  G PRECN3  44E4  G PRENF = 009C
PS    = 00B2    PSTAT = 0048    PSW   =%0009    PTSTOK  44A7  G
PTY     448A    PTYC    44C8    PTY1    4495    PTY2    44A2
PXPTBL  4664    P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020
P.INTE= 0080    P.LCS = 0040    P.LWR = 0020    P.RDP = 0002
P.RPEN= 0004    P.RPST= 0002    P.RP0E= 0020    P.RP1E= 0010
P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080    P.STAT= 0002
P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004
P.WDS = 0040    P.WFLP= 0001    P.WPEN= 0010    P.WP0E= 0008
P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002
QUE   = 281B    QUEM  = 281E    RAMT  = 0010    RARA  = 0006
RARAI = 0004    RCHBD0= 0048    RCHBD1= 0047    RCHOK = 0046
RCHTST= 000C    RCLRT = 000D    RCMD  = 000B    RCMLP = 0003
RCONT = 0080    RDATA = 0017    RDCLK = 0010    RDON  = 0011
READG = 0007    RECCN1  451A  G RECCN2  4543  G RECCN3  4535  G
RECEND  44FE    REND  = 0014    REQST = 2806    RESCHR= 00D1
REVTST= 0064    REWIND= 0004    RFIFOL= 0008    RGCLK = 0002
RGCRI = 0003    RIBG  = 0001    RILL  = 0012    RINST = 000C
RMCTST= 0008    RMK2  = 0013    RNOP  = 0000    RPATH = 0001
RPBAD = 0044    RPCHI = 0001    RPCLK = 0003    RPCTL = 0009
RPEI  = 0002    RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E
RPOK  = 0043    RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A
RSTAT = 0002    RTIEB = 000A    RTIER = 0030    RTM   = 0005
RUNKI = 0009    RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001
R.BOP = 0008    R.DATA= 0040    R.DON = 0002    R.DRDY= 0010
R.END = 0010    R.ILL = 0004    R.JVOK= 0004    R.MK2 = 0008
```

```
R.PLOD= 0008    R.PLO0= 0010    R.PLO1= 0020    R.POST= 0020
R.STNM= 0002    R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080
R.TSTD= 0040    R.VOK = 0080    R00H  = 0081    R00L  = 0080
R01H  = 0083    R01L  = 0082    R02H  = 0085    R02L  = 0084
R03H  = 0087    R03L  = 0086    R04H  = 0089    R04L  = 0088
R05H  = 008B    R05L  = 008A    R06H  = 008D    R06L  = 008C
R07H  = 008F    R07L  = 008E    R10H  = 0091    R10L  = 0090
R11H  = 0093    R11L  = 0092    R12H  = 0095    R12L  = 0094
R13H  = 0097    R13L  = 0096    R14H  = 0099    R14L  = 0098
R15H  = 009B    R15L  = 009A    R16H  = 009D    R16L  = 009C
R17H  = 009F    R17L  = 009E    R7.5  = 0010    SELCLR= 0000
SEQERR  453E    SETATA= 00A1    SID   = 0080    SOD   = 0080
SOE   = 0040    SP    =%0008    SSCLK = 0040    SSTEP = 0005
STACK = 4FFF    STATRM= 4F20    STPCT   4F20    STRSP = 5000
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TMF   = 0099    TMRDY = 0040
TRANSC  4694    TRKENA= 00D2    TSET  = 2803    TSTEND= 2818
TSTS  = 0040    TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010
T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002
T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020    T.PES = 0008
T.PSBJ= 0020    T.PS0J= 0008    T.PS1J= 0010    T.RDY = 0080
T.RDY0= 0040    T.RWD = 0010    T.SCLK= 0002    T1LOOP  4305
T1LPA   4330    UIBG  = 00A1    UNITMP  468D    VALFC   4F98
VALTB   4F95    VELTST= 005B    WDR.P = 0010    WMCCTL= 00D3
WMCERR= 00DA    WMCSTA= 00D0    WRTCLK= 0000    WRTDAT= 00D3
W.ACRC= 0004    W.CRC = 0008    W.DIAG= 0002    W.DONN= 0040
W.ECC = 0010    W.ENAB= 0080    W.ERR = 0020    W.FMT = 0070
W.GCR = 0010    W.LEFT= 0004    W.ONES= 0020    W.RESI= 0002
W.REV = 0004    W.ROME= 0010    W.RST = 0001    W.SKIP= 000F
W.WRIT= 0008    W.XFER= 0020    X     =%000A    XDATA   468E
XDLOOP  442B    XDLPC   4446    XDLPC1  4410    XDLPC2  441D
XDLP1   4438    XMEND   4549    XPLOOP  44B0    XPLP1   44BD
X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002    X.ROME= 0001
X.WCLK= 0001    Y     -%0008    .     = 4697


 ERRORS DETECTED:  0

 *XMC3.A78/PTP,XMC3-NLIST,PARAM,MACRO,LIST,XMC3
 RUN-TIME: 4 5 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:10 SEQ 0485
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  XMC4 - TRANSLATOR MICRO CONTROLLER PART #4
                                .SBTTL  TEST 1 - XMC - GCR - PREAMBLE/POSTAMBLE TEST NO DATA GROUP
                                ;ID     XMC4-TRANSLATOR MICRO CONTROLLER PART #4
1332  4300                      ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR - PREAMBLE/POSTAMBLE TEST - NO DATA GROUP
1334  4300                      SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A GCR RECORD OF ONE BYTE
                                ;AND THEN SINGLE STEPS THE WRITE PATH (WMC, XMC) WHILE MONITORING THE
                                ;XMC ''XL WCLK'' BIT.  THE OUTPUT OF THE TRANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, END MARK, RESIDUAL GROUP, CRC GROUP, AND POSTAMBLE.
                                ;THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT, DATA STATES AND FLUX REVERSAL RATE.
1340  4300                      SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    INITIALIZE THE BYTE COUNTER, PAD COUNTER, AND ERROR COUNTER
                                ;*    CLEAR THE ERROR COUNTER
                                ;*    REQUEST A TU PORT FROM THE HOST
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    IF NO UNITS SELECTED
                                ;*    :    THEN-ERROR-ABORT TEST
                                ;*    :    ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    SET LOOP COMMAND/STATUS ON SELECTED PORT
                                ;*    CLEAR THE COMMAND BYTE FOR SELECTED PORT
                                ;*    SET WRITE PATH ENABLE
                                ;*    INITIALIZE THE EXPECTED RESULTS TABLE POINTER
                                ;*    SET WMC/XMC RESTART
                                ;*    SELECT THE SINGLE STEP CLOCK
                                ;*    CLEAR WMC/XMC RESTART
                                ;*    LOAD THE BYTE COUNT WITH 1 (BITS 7:0)
                                ;*    CLEAR THE BYTE COUNT (BITS 15:8)
                                ;*    SET DDR CONTROL TO ''IN''
                                ;*    SELECT THE IMAGE MODE OF DATA FORMATTING
                                ;*    LOAD THE RESIDUAL CHARACTER WITH 40(8)
                                ;*    LOAD THE PAD COUNT WITH 6 (BITS 7:0)
                                ;*    CLEAR THE PAD COUNT (BITS 15:8)
                                ;*    ENABLE ALL TRACKS LOAD TRACK ENABLE WITH ONES
                                ;*    SET THE CLOCK COUNT TO 180(10)
                                ;*    BGNDO
                                ;*    :    CALL SUBROUTINE CLKSYS
                                ;*    :    DECREMENT THE CLOCK COUNT
                                ;*    :    DO UNTIL CLOCK COUNT=0
                                ;*    ENDDO
                                ;*    SET WRITE MICRO TO WRITE MICRO ENABLE, TRANSLATOR ENABLE, GCR, AND WRITE
                                ;*    SET THE CLOCK COUNT TO 256(10)
                                ;*    BGNDO
```

```
1374                    ;*      :    CALL SUBROUTINE CLKSYS
1375                    ;*      :    DECREMENT THE CLOCK COUNT
1376                    ;*      :    DO UNTIL CLOCK COUNT=0 OR XL WRITE CLOCK=1
1377                    ;*      ENDDO
1378                    ;*      IF CLOCK COUNT=0
1379                    ;*      :    THEN-FATAL ERROR-ABORT TEST
1380                    ;*      :    ELSE-CONTINUE
1381                    ;*      ENDIF
1382                    ;*      BGNDO
1383                    ;*      :    CALL SUBROUTINE CLKSYS
1384                    ;*      :    CALL SUBROUTINE CLKSYS
1385                    ;*      :    INPUT THE TRANSLATOR OUTPUT DATA
1386                    ;*      :    IF THE ACTUAL TRANSLATOR OUTPUT=EXPECTED TRANSLATOR OUTPUT
1387                    ;*      :    :    THEN-CONTINUE
1388                    ;*      :    :    ELSE-ERROR
1389                    ;*      :    ENDIF
1390                    ;*      :    INPUT THE TRANSLATOR OUTPUT DATA PARITY
1391                    ;*      :    IF DATA OUTPUT DURING DATA GROUP
1392                    ;*      :    :    THEN-EXPECTED PARITY=ODD PARITY FOR OUTPUT DATA
1393                    ;*      :    :    ELSE-EXPECTED PARITY=1
1394                    ;*      :    ENDIF
1395                    ;*      :    IF INPUT PARITY=EXPECTED PARITY
1396                    ;*      :    :    THEN-CONTINUE
1397                    ;*      :    :    ELSE-ERROR
1398                    ;*      :    ENDIF
1399                    ;*      :    SET THE LOOP COUNT TO 14(8)
1400                    ;*      :    BGNDO
1401                    ;*      :    :    CALL SUBROUTINE CLKSYS
1402                    ;*      :    :    DECREMENT THE LOOP COUNT
1403                    ;*      :    :    DO UNTIL THE LOOP COUNT = 0
1404                    ;*      :    ENDDO
1405                    ;*      :    IF XL WRITE CLOCK = 1
1406                    ;*      :    :    THEN - CONTINUE
1407                    ;*      :    :    ELSE - ERROR - IMPROPER TRANSITION RATE FOR GCR
1408                    ;*      :    ENDIF
1409                    ;*      :    INCREMENT THE EXPECTED RESULTS TABLE POINTER
1410                    ;*      :    DO UNTIL THE EXPECTED RESULTS POINTER=END OF TABLE
1411                    ;*      ENDDO
1412                    ;*      SET THE CLOCK COUNT TO 256(10)
1413                    ;*      BGNDO
1414                    ;*      :    CALL SUBROUTINE CLKSYS
1415                    ;*      :    DECREMENT THE CLOCK COUNT
1416                    ;*      :    DO UNTIL CLOCK COUNT=0 OR TRANSLATOR OR WRITE MICRO DONE 0
1417                    ;*      ENDDO
1418                    ;*      IF CLOCK COUNT=0
1419                    ;*      :    THEN-ERROR-TIMEOUT WAITING FOR WRITE MICRO DONE
1420                    ;*      :    ELSE-CONTINUE
1421                    ;*      ENDIF
1422                    ;*      IF TRANSLATOR DONE
1423                    ;*      :    THEN-ERROR-TRANSLATOR DONE BEFORE WRITE MICRO
1424                    ;*      :    ELSE-SET THE CLOCK COUNT TO 256(10)
1425                    ;*      :    BGNDO
1426                    ;*      :    :    CALL SUBROUTINE CLKSYS
1427                    ;*      :    :    DECREMENT THE CLOCK COUNT
```

```
                                ;*     :     :    DO UNTIL TRANSLATOR DONE OR CLOCK COUNT=0
                                ;*     :     ENDDO
                                ;*     :     IF CLOCK COUNT=0
                                ;*     :     :    THEN-ERROR-TIMEOUT WAITING FOR TRANSLATOR DONE
                                ;*     :     :    ELSE-CONTINUE
                                ;*     :     ENDIF
                                ;*    ENDIF
                                ;*ENDTST
1436  4300                      SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1437                            ;*XMC4 MICRO TEST 01
1438                            ;*XMC4 MICRO ERROR 01
1439                            ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1440                            ;*M8958, M8959, M8955'S
1441                            ;*OPERATOR ERROR NO TM78 UNITS SELECTED
1442                            ;*FATAL ERROR - TEST ABORTED
1443                            ;*
1444                            ;*XMC4 MICRO TEST 01
1445                            ;*XMC4 MICRO ERROR 02
1446                            ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1447                            ;*M8958, M8959, M8955'S
1448                            ;*WRITE PATH TIMEOUT WHILE WAITING FOR 'XL CLK'' TO SET
1449                            ;*FATAL ERROR - TEST ABORTED
1450                            ;*
1451                            ;*XMC4 MICRO TEST 01
1452                            ;*XMC4 MICRO ERROR 03
1453                            ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1454                            ;*M8958, M8959, M8955'S
1455                            ;*DATA STATE RECEIVED FROM THE XMC INCORRECT
1456                            ;*ACTUAL = NNNN
1457                            ;*EXPECTED = NNNN
1458                            ;*TRANSITION COUNT = LLL
1459                            ;*
1460                            ;*XMC4 MICRO TEST 01
1461                            ;*XMC4 MICRO ERROR 04
1462                            ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1463                            ;*M8958, M8959, M8955'S
1464                            ;*PARITY STATE RECEIVED FROM THE XMC INCORRECT
1465                            ;*ACTUAL = NNNN
1466                            ;*EXPECTED = NNNN
1467                            ;*TRANSITION COUNT = LLL
1468                            ;*
1469                            ;*XMC4 MICRO TEST 01
1470                            ;*XMC4 MICRO ERROR 05
1471                            ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1472                            ;*M8958, M8959, M8955'S
1473                            ;*TRANSLATOR 'XL WCLK'' NOT SET AFTER EIGHT CLOCK CYCLES
1474                            ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1475                            ;*TRANSITION COUNT = LLL
1476                            ;*
1477                            ;*XMC4 MICRO TEST 01
1478                            ;*XMC4 MICRO ERROR 06
```

```
                                        ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
                                        ;*M8958, M8959, M8955'S
                                        ;*''WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
                                        ;*
                                        ;*XMC4 MICRO TEST 01
                                        ;*XMC4 MICRO ERROR 07
                                        ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
                                        ;*M8958, M8959, M8955'S
                                        ;*''XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
                                        ;*
                                        ;*XMC4 MICRO TEST 01
                                        ;*XMC4 MICRO ERROR 10
                                        ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
                                        ;*M8958, M8959, M8955'S
                                        ;*''XMC NOT DONE'' RESET BEFORE 'WMC NOT DONE'' RESET
                                        ;
4300                                    ; ***************************************************************************
                                        ;
4300                                    TEST1:  TESTX   @1
4300  3E  01              7.0                       MVI     A,@1            ;DEFINE THE TEST NUMBER
4302  CD  03  28         18.0                       CALL    TSET            ;SETUP THE TEST
                                        ;%XMC4-GCR-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
                                        ;&M8958, M8959, M8955'S
                                        ;
4305  AF                 4.0          XRA     A               ;CLEAR ALL TRACK ENABLE LINES
4306  D3  D2              10.0          OUT     TRKENA
4308  D3  D2              10.0          OUT     TRKENA
430A  AF                 4.0          XRA     A               ;CLEAR UNIT MAP STORAGE
430B  32  7E  46           13.0          STA     UNITMP          ;1ST TIME HERE
430E  32  82  46           13.0          STA     EXDLPC          ;CLEAR THE ERROR COUNTER
                                        ;
4311  3E  30                7.0  T1LOOP: MVI     A,@60           ;INIT THE BYTE COUNTER
4313  D3  D7              10.0          OUT     CNTCTL          ;TO NORMAL MODE
4315  3E  70                7.0          MVI     A,@160          ;INIT THE PAD COUNTER
4317  D3  D7              10.0          OUT     CNTCTL          ;TO NORMAL MODE
4319  3E  B0                7.0          MVI     A,@260          ;INIT THE ERROR COUNTER
431B  D3  D7              10.0          OUT     CNTCTL          ;TO NORMAL MODE
431D                                    CLRECT
431D  AF                 4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
431E  D3  D6              10.0                OUT     ERRCNT          ;CLEAR BITS 7-0
4320  D3  D6              10.0                OUT     ERRCNT          ;CLEAR BITS 15-8
                                        ;
                                        ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
                                        ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.
                                        ;
4322  3A  7E  46           13.0          LDA     UNITMP          ;GET THE UNIT MAP STORAGE
4325  A7                 4.0          ANA     A
4326  C2  3F  43           10.0          JNZ     T1LPA           ;JUMP OVER IF ALREADY HAVE IT
4329                                    REQ     @7,0,0,0
4329  CD  06  28         18.0                 CALL    REQST
432C  00                                   .BYTE   0               ;DATA PATTERN NUMBER
432D  00  00                               .WORD   0               ;SYSTEM ''0'' COUNT
432F  00  00                               .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
4331  00                                   .BYTE                   ;DATA COMPARE FLAG IF =1
```

```
  (1)  4332  07                                          .BYTE   @7              ;REQUEST CODE
1522   4333                                      RIN     R12L
  (1)  4333  DB  94          10.0                        IN      R12L            ;READ R12L INTO AC
  (1)  4335  7F               4.0                        MOV     A,A             ;RETRY LINK
1523   4336  32  7E  46       13.0                STA     UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
1524   4339  AF               4.0                XRA     A
1525   433A  CD  76  45       18.0                CALL    CLEAR
1526   433D  06  00            7.0                MVI     B,0
1527   433F  3A  7E  46       13.0  T1LPA:       LDA     UNITMP          ;GET SELECTED UNIT MAP
1528   4342  E6  01            7.0                ANI     $01             ;UNIT 0 PRESENT?
1529   4344  C2  70  43       10.0                JNZ     FOUND           ;YES-USE IT
1530   4347  04               4.0                INR     B
1531   4348  3A  7E  46       13.0                LDA     UNITMP          ;
1532   434B  E6  02            7.0                ANI     $02             ;UNIT 1 PRESENT?
1533   434D  C2  70  43       10.0                JNZ     FOUND           ;YES-USE IT
1534   4350  04               4.0                INR     B
1535   4351  3A  7E  46       13.0                LDA     UNITMP          ;
1536   4354  E6  04            7.0                ANI     $04             ;UNIT 2 PRESENT?
1537   4356  C2  70  43       10.0                JNZ     FOUND           ;YES-USE IT
1538   4359  04               4.0                INR     B
1539   435A  3A  7E  46       13.0                LDA     UNITMP
1540   435D  E6  08            7.0                ANI     $08             ;UNIT 3 PRESENT
1541   435F  C2  70  43       10.0                JNZ     FOUND           ;YES-USE IT
1542                                                                      ;ELSE - OPERATOR ERROR
1543   4362                                      ERR     EXIT,DUMMY
  (1)                                             ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4362  CD  09  28       18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)        0001                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4365  01                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4366  00                                           .BYTE
  (1)  4367  CD  15  28       18.0          DUMMY::   CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  436A  DA  67  45       10.0                    JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1544                                        ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1545                                        ;>FATAL ERROR - TEST ABORTED
1546   436D  C3  67  45       10.0                JMP     EXIT
1547
1548                                        ;SET UP FOR THE DATA TRANSFER
1549   4370  DB  E0            10.0  FOUND:  IN      INTSTA
1550   4372  E6  80             7.0          ANI     BIT7
1551   4374  B0               4.0          ORA     B
1552   4375  D3  E0            10.0          OUT     MBSEL
1553   4377  3E  40             7.0          MVI     A,P.LCS         ;KILL THE MTA BOARD
1554   4379  D3  48            10.0          OUT     PDIAG
1555   437B  AF               4.0          XRA     A
1556   437C  D3  40            10.0          OUT     TCMD
1557
1558   437E  3E  10             7.0          MVI     A,P.WPEN
1559   4380  D3  4C            10.0          OUT     PENAB           ;ENABLE THE WRITE PATH
1560   4382  21  A1  45       10.0          LXI     H,GXPTBL        ;LOAD ADDRESS OF EXPECTED
1561   4385  22  83  46       16.0          SHLD    POINTER         ;RESULTS TABLE INTO POINTER
1562   4388  3E  01             7.0          MVI     A,W.RST         ;LOAD THE WMC/XMC RESTART CODE
1563   438A  D3  D3            10.0          OUT     WMCCTL          ;ISSUE THE COMMAND
1564   438C  3E  05             7.0          MVI     A,SSTEP         ;SELECT SINGLE STEP WRITE CLOCK
1565   438E  D3  F0            10.0          OUT     CLKCTL          ;INITIALIZE THE CLOCK
```

```
1566  4390  32  87  46    13.0          STA     CCTLWD          ;SAVE SOFTWARE IMAGE
1567  4393  AF              4.0          XRA     A               ;CLEAR THE ACC.
1568  4394  D3  D3          10.0         OUT     WMCCTL          ;CLEAR THE WMC/XMC RESTART BIT
1569  4396  3E  01          7.0          MVI     A,1             ;GET THE BYTE COUNT - NO DATA GROUP
1570  4398  D3  D4          10.0         OUT     BYTCNT          ;LOAD BITS 7-0
1571  439A  AF              4.0          XRA     A
1572  439B  D3  D4          10.0         OUT     BYTCNT          ;CLEAR THE BYTE COUNTER BITS 15-8
1573  439D  32  85  46      13.0         STA     TRANSCNT        ;INITIALIZE THE TRANSITION COUNT
1574  43A0  32  86  46      13.0         STA     TRANSCNT+1
1575  43A3  32  80  46      13.0         STA     DPCNT           ;CLEAR THE PARITY TEST COUNTER
1576  43A6  3E  88          7.0          MVI     A,@210
1577  43A8  D3  DB          10.0         OUT     DDRCTL          ;SET THE DDR CONTROL TO 'IN'
1578  43AA  3E  50          7.0          MVI     A,$50           ;LOAD THE IMAGE MODE COMMAND
1579  43AC  D3  D0          10.0         OUT     DATACTL         ;SET THE DATA CONTROL MODE
1580  43AE  3E  20          7.0          MVI     A,@40           ;LOAD THE RESIDUAL CHARACTER VALUE
1581                                                             ;1 DATA BYTE IN RESID GROUP
1582  43B0  D3  D1          10.0         OUT     RESCHR          ;SET THE RESIDUAL CHARACTER
1583  43B2  3E  05          7.0          MVI     A,5             ;LOAD PAD COUNT VALUE
1584  43B4  3C              4.0          INR     A               ;CORRECT FOR THE INTEL 8253 CHIP
1585  43B5  D3  D5          10.0         OUT     PADCNT          ;WRITE TO WMC BITS 7-0
1586  43B7  AF              4.0          XRA     A               ;LOAD HI VALUE
1587  43B8  D3  D5          10.0         OUT     PADCNT          ;WRITE TO WMC
1588  43BA  3E  FF          7.0          MVI     A,$FF           ;LOAD ACC WITH ALL ONES
1589  43BC  D3  D2          10.0         OUT     TRKENA          ;ENABLE PARITY TRACK
1590  43BE  D3  D2          10.0         OUT     TRKENA          ;ENABLE DATA TRACKS
1591  43C0  06  B4          7.0          MVI     B,180           ;LOAD THE CLOCK COUNT
1592
1593  43C2  CD  6A  45      18.0  CLKRS:  CALL    CLKSYS          ;CLOCK THE THING
1594  43C5  05              4.0          DCR     B               ;
1595  43C6  C2  C2  43      10.0         JNZ     CLKRS           ;CONTINUE WITH RESTART CLOCKING
1596
1597  43C9  3E  D8          7.0          MVI     A,W.ENAB!X.ENAB!W.GCR!W.WRITE
1598  43CB  D3  D3          10.0         OUT     WMCCTL          ;START THE OPERATION
1599  43CD  06  FF          7.0  CLK1:   MVI     B,$FF           ;SET TIMEOUT COUNT TO 256
1600                                                             ;
1601  43CF  CD  6A  45      18.0  CLK:    CALL    CLKSYS          ;SINGLE STEP THE WRITE PATH
1602                                                             ;
1603  43D2  DB  D0          10.0         IN      WMCSTA          ;GET THE WMC STATUS
1604  43D4  E6  01          7.0          ANI     X.WCLK          ;X.WCLK SET?
1605  43D6  C2  EB  43      10.0         JNZ     BGNPRE          ;YES-BEGINNING OF PREAMBLE
1606  43D9  05              4.0          DCR     B               ;NO-DECREMENT CLOCK
1607  43DA  C2  CF  43      10.0         JNZ     CLK             ;CONTINUE UNTIL ZERO
1608  43DD                               ERR     TSTEND,ABOCN0
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43DD  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E0  02                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E1  00                                 .BYTE
 (1)  43E2  CD  15  28      18.0         ABOCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E5  DA  18  28      10.0                 JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1609                                   ;>WRITE PATH TIMEOUT WHILE WAITING FOR 'XL CLK'' TO SET
1610                                   ;>FATAL ERROR - TEST ABORTED
1611
1612  43E8  C3  18  28      10.0         JMP     TSTEND
```

```
1613
1614  43EB  CD 6A 45   18.0  BGNPRE: CALL   CLKSYS          ;CLOCK THE SYSTEM
1615  43EE  CD 6A 45   18.0          CALL   CLKSYS          ;CLOCK AGAIN
1616  43F1  DB 40      10.0          IN     TSTS            ;GET THE DATA READ
1617  43F3  47           4.0          MOV    B,A             ;SAVE IT IN B=ADATA
1618  43F4                            ROUT   ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  43F4  D3 94        10.0                 OUT    ADATA           ;WRITE AC INTO ADATA
 (1)  43F6  7F           4.0                  MOV    A,A             ;RETRY LINK
1619                                                       ;
1620  43F7  2A 85 46   16.0          LHLD   TRANSCNT        ;GET THE TRANSITION COUNT
1621  43FA  23           6.0          INX    H               ;UPDATE
1622  43FB  22 85 46   16.0          SHLD   TRANSCNT        ;RESTORE THE TRANSITION COUNT
1623  43FE  7D           4.0          MOV    A,L             ;GET TRANSCOUNT LOW
1624  43FF                            ROUT   R05L            ;WRITE TO CAS
 (1)  43FF  D3 8A        10.0                 OUT    R05L            ;WRITE AC INTO R05L
 (1)  4401  7F           4.0                  MOV    A,A             ;RETRY LINK
1625  4402  7C           4.0          MOV    A,H             ;GET TRANSCOUNT HIGH
1626  4403                            ROUT   R05H            ;WRITE TO CAS
 (1)  4403  D3 8B        10.0                 OUT    R05H            ;WRITE AC INTO R05H
 (1)  4405  7F           4.0                  MOV    A,A             ;RETRY LINK
1627  4406  2A 83 46   16.0          LHLD   POINTER         ;GET POINTER TO EXPECTED DATA
1628  4409  7E           7.0          MOV    A,M             ;GET EXPECTED DATA BYTE
1629  440A  32 7F 46   13.0          STA    XDATA           ;SAVE FOR LATER
1630  440D  B8           4.0          CMP    B               ;COMPARE WITH ACTUAL
1631  440E  CA 26 44   10.0          JZ     PRECN1          ;CONTINUE IF EQUAL
1632                                                       ;ELSE-ERROR
1633  4411                            ROUT   EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  4411  D3 95        10.0                 OUT    EDATA           ;WRITE AC INTO EDATA
 (1)  4413  7F           4.0                  MOV    A,A             ;RETRY LINK
1634  4414  3A 82 46   13.0  XDLPC1: LDA    EXDLPC          ;GET THE CURRENT COUNT AT ERROR
1635  4417  A7           4.0          ANA    A
1636  4418  C2 21 44   10.0          JNZ    XDLPC2          ;STAY AT THIS COUNT IF NOT ZERO
1637  441B  3A 85 46   13.0          LDA    TRANSCNT        ;GET THE COUNT AT THE ERROR
1638  441E  32 82 46   13.0          STA    EXDLPC          ;SAVE IT
1639  442'                    XDLPC2: ERRB   XDLOOP,PRECN1,@4
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44.    CD 12 28    18.0                       CALL   ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       )003                                    MSGN   =      MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4424  03                                      .BYTE  MSGN            ;MESSAGE NUMBER ID
 (1)  4425  04                                      .BYTE  @4              ;PRINT ROUTINE NUMBER
 (1)  4426  CD 15 28    18.0                PRECN1::       CALL   CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4429  DA 2F 44    10.0                       JC     XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1640                            ;>DATA STATE RECEIVED FROM THE XMC INCORRECT
1641  442C  C3 4A 44   10.0          JMP    XDLPC           ;SKIP OVER ERROR LOOP CONTROL
1642
1643
1644
1645                            ;HERE IS THE ERROR LOOP CONTROL
1646
1647  442F  3A 94 4F   13.0  XDLOOP: LDA    LPFLG           ;GET THE LOOP ON ERROR FLAG
1648  4432  A7           4.0          ANA    A
1649  4433  C2 3C 44   10.0          JNZ    XDLP1           ;JUMP IF FLAG IS SET
1650  4436  32 82 46   13.0          STA    EXDLPC          ;ELSE - CLEAR THE COUNTER
1651  4439  C3 4A 44   10.0          JMP    XDLPC           ;AND CONTINUE TESTING
```

```
1652
1653  443C  3A 85 46   13.0  XDLP1:  LDA   TRANSCNT        ;GET THE LOW BYTE TRANSITION COUNT
1654  443F  47           4.0         MOV   B,A             ;TEMP SAVE
1655  4440  3A 82 46   13.0         LDA   EXDLPC          ;GET THE TRANS. COUNT AT ERROR
1656  4443  B8           4.0         CMP   B               ;SAME?
1657  4444  CA 11 43   10.0         JZ    T1LOOP          ;YUP - LOOP NOW!
1658  4447  C3 4A 44   10.0         JMP   XDLPC           ;NO - CONTINUE TILL A MATCH...
1659
1660  444A  DB 48        10.0  XDLPC:  IN    PSTAT           ;GET THE PORT ADDRESS
1661  444C  0F           4.0         RRC                   ;RIGHT JUSTIFY
1662  444D  0F           4.0         RRC                   ;
1663  444E  E6 01        7.0         ANI   $01             ;REMOVE OTHER BITS
1664  4450  57           4.0         MOV   D,A             ;STORE ACTUAL PARITY IN REG D
1665
1666  4451                           ROUT  ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  4451  D3 94        10.0               OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  4453  7F           4.0                MOV   A,A             ;RETRY LINK
1667  4454  2A 83 46   16.0         LHLD  POINTER
1668  4457  01 FB 45   10.0         LXI   B,GXPDAT        ;LOAD BEGINNING ADDRESS OF EXPECTED DATA
1669  445A  78           4.0         MOV   A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1670  445B  2F           4.0         CMA                   ;PAIR B & C
1671  445C  47           4.0         MOV   B,A
1672  445D  79           4.0         MOV   A,C
1673  445E  2F           4.0         CMA
1674  445F  4F           4.0         MOV   C,A
1675  4460  03           6.0         INX   B
1676  4461  09          10.0         DAD   B               ;ADD POINTER AND BEGINNING OF DATA TABLE
1677  4462  D2 89 44   10.0         JNC   PPREAM          ;CHECK PREAMBLE PARITY TRACK
1678
1679  4465  2A 83 46   16.0         LHLD  POINTER         ;RESET H & L REGS
1680  4468  01 14 46   10.0         LXI   B,GDATEN        ;GET THE ENDING ADDRESS OF EXPECTED DATA
1681  446B  78           4.0         MOV   A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1682  446C  2F           4.0         CMA                   ;PAIR B & C
1683  446D  47           4.0         MOV   B,A
1684  446E  79           4.0         MOV   A,C
1685  446F  2F           4.0         CMA
1686  4470  4F           4.0         MOV   C,A
1687  4471  03           6.0         INX   B
1688  4472  09          10.0         DAD   B               ;ADD POINTER AND END OF DATA TABLE
1689  4473  DA 89 44   10.0         JC    PPOST           ;CHECK POSTAMBLE PARITY TRACK
1690
1691  4476  01 64 46   10.0  DPTST:  LXI   B,PXPTBL        ;POINT TO EXPECTED PARITY DATA TABLE
1692  4479  2A 80 46   16.0         LHLD  DPCNT           ;POINT TO THE TABLE OFFSET NUMBER
1693  447C  23           6.0         INX   H               ;UPDATE IT
1694  447D  22 80 46   16.0         SHLD  DPCNT           ;SAVE NEW COUNT
1695  4480  09          10.0         DAD   B               ;MAKE REAL POINTER
1696  4481  7E           7.0         MOV   A,M             ;GET EXPECTED PARITY STATE
1697  4482  A7           4.0         ANA   A               ;SET CONDITION BITS
1698  4483  FA D2 44   10.0         JM    PTYC            ;JMP IF NO COMPARE ON THIS DATA
1699  4486  C3 91 44   10.0         JMP   PTY             ;TEST THE PARITY BIT
1700
1701                          ;HERE TO CHECK THE PREAMBLE/POSTAMBLE PARITY TRACK BIT
1702
1703  4489                    PPOST:
```

```
1704  4489  3A  7F  46      13.0  PPREAM: LDA   XDATA           ;GET THE DATA EXPECTED
1705  448C  E6  01              7.0         ANI   1               ;PARITY TRACK IS SAME
1706  448E  C3  91  44        10.0         JMP   PTY             ;CHECK THE PARITY
1707
1708  4491  BA               4.0  PTY:    CMP   D               ;SEE IF EXPECTED=ACTUAL
1709  4492  CA  AE  44        10.0         JZ    PTSTOK          ;JMP IF OK
1710  4495                             ROUT  EDATA           ;SAVE THE EXPECTED DATA
 (1)  4495  D3  95              10.0                 OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  4497  7F                4.0                 MOV   A,A             ;RETRY LINK
1711  4498  7A                4.0         MOV   A,D             ;ERROR - GET THE ACTUAL DATA
1712  4499                             ROUT  ADATA           ;SAVE FOR PRINTOUT
 (1)  4499  D3  94              10.0                 OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  449B  7F                4.0                 MOV   A,A             ;RETRY LINK
1713
1714  449C  3A  82  46        13.0  PTY1:   LDA   EXDLPC          ;GET THE COUNT AT THE ERROR
1715  449F  A7                4.0         ANA   A
1716  44A0  C2  A9  44        10.0         JNZ   PTY2            ;JUMP IF AT AN ERROR NOW
1717  44A3  3A  85  46        13.0         LDA   TRANSCNT        ;GET THE COUNT AT ERROR
1718  44A6  32  82  46        13.0         STA   EXDLPC          ;SAVE IT
1719  44A9                       PTY2:   ERRB  XPLOOP,PTSTOK,@4
 (1)                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  44A9  CD  12  28        18.0                 CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                                    MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44AC  04                                    .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44AD  04                                    .BYTE @4              ;PRINT ROUTINE NUMBER
 (1)  44AE  CD  15  28        18.0           PTSTOK::        CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44B1  DA  B7  44        10.0                 JC    XPLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1720                        ;>PARITY STATE RECEIVED FROM THE XMC INCORRECT
1721  44B4  C3  D2  44        10.0         JMP   PTYC            ;JMP OVER CODE
1722
1723  44B7  3A  94  4F        13.0  XPLOOP: LDA   LPFLG           ;GET THE LOOP ON ERROR FLAG
1724  44BA  A7                4.0         ANA   A
1725  44BB  C2  C4  44        10.0         JNZ   XPLP1           ;JUMP IF FLAG IS SET
1726  44BE  32  82  46        13.0         STA   EXDLPC          ;ELSE - CLEAR THE COUNTER
1727  44C1  C3  D2  44        10.0         JMP   PTYC            ;AND CONTINUE TESTING
1728
1729  44C4  3A  85  46        13.0  XPLP1:  LDA   TRANSCNT        ;GET THE LOW BYTE TRANSITION COUNT
1730  44C7  47                4.0         MOV   B,A             ;TEMP SAVE
1731  44C8  3A  82  46        13.0         LDA   EXDLPC          ;GET THE TRANS. COUNT AT ERROR
1732  44CB  B8                4.0         CMP   B               ;SAME?
1733  44CC  CA  11  43        10.0         JZ    T1LOOP          ;YUP - LOOP NOW!
1734  44CF  C3  D2  44        10.0         JMP   PTYC            ;NO - CONTINUE TILL A MATCH...
1735
1736  44D2  2A  83  46        16.0  PTYC:   LHLD  POINTER         ;NO-UPDATE THE
1737  44D5  23                6.0         INX   H               ;TABLE POINTER
1738  44D6  22  83  46        16.0         SHLD  POINTER         ;ADDRESS
1739
1740  44D9  06  0E              7.0         MVI   B,14            ;LOAD THE CLOCK COUNT
1741  44DB  CD  6A  45        18.0  1$:     CALL  CLKSYS          ;CLOCK THE WRITE PATH
1742  44DE  05                4.0         DCR   B               ;DECREMENT THE COUNT
1743  44DF  C2  DB  44        10.0         JNZ   1$              ;CONTINUE UNTIL 8 CLOCKS
1744  44E2  DB  D0            10 0         IN    WMCSTA
1745  44E4  E6  01              7.0         ANI   X.WCLK
1746  44E6  C2  EE  44        10.0         JNZ   PRECN3
```

```
1747  44E9                                          ERR     T1LOOP,PRECN3,@4
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E9  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44EC  05                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44ED  04                                              .BYTE   @4
 (1)  44EE  CD  15  28        18.0                  PRECN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44F1  DA  11  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1748                                    ;>TRANSLATOR 'XL WCLK'' NOT SET AFTER EIGHT CLOCK CYCLES
1749                                    ;>IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1750
1751  44F4  2A  83  46        16.0            LHLD    POINTER         ;GET THE TABLE POINTER
1752  44F7  01  63  46        10.0            LXI     B,GTBLEN        ;GET END OF TABLE ADDRESS
1753  44FA  78                 4.0            MOV     A,B             ;FROM THE
1754  44FB  2F                 4.0            CMA                     ;TWO'S COMPLIMENT
1755  44FC  47                 4.0            MOV     B,A             ;OF THE TABLE END
1756  44FD  79                 4.0            MOV     A,C             ;ADDRESS
1757  44FE  2F                 4.0            CMA
1758  44FF  4F                 4.0            MOV     C,A
1759  4500  03                 6.0            INX     B
1760  4501  09                10.0            DAD     B               ;END OF TABLE?
1761  4502  DA  08  45        10.0            JC      RECEND          ;YES
1762  4505  C3  EB  43        10.0            JMP     BGNPRE
1763
1764  4508  06  FF             7.0  RECEND: MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1765  450A  CD  6A  45        18.0  1$:     CALL    CLKSYS          ;CLOCK THE SYSTEM
1766  450D  DB  D0            10.0          IN      WMCSTA          ;INPUT THE WMC STATUS
1767  450F  E6  40             7.0          ANI     W.DONN          ;WMC DONE?
1768  4511  CA  24  45        10.0          JZ      RECCN1          ;YES-EXIT
1769  4514  DB  D0            10.0          IN      WMCSTA          ;GET WMC STATUS
1770  4516  E6  80             7.0          ANI     X.DONN          ;XMC DONE
1771  4518  CA  48  45        10.0          JZ      SEQERR          ;YES-ERROR
1772  451B  05                 4.0          DCR     B               ;NO-DECREMENT THE COUNT
1773  451C  C2  0A  45        10.0          JNZ     1$              ;CONTINUE IF NOT ZERO
1774  451F                                  ERR     RECEND,RECCN1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  451F  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4522  06                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4523  00                                              .BYTE
 (1)  4524  CD  15  28        18.0                  RECCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4527  DA  08  45        10.0                          JC      RECEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1775                                    ;>'WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1776
1777  452A  06  FF             7.0          MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1778  452C  CD  6A  45        18.0  3$:     CALL    CLKSYS          ;CLOCK THE SYSTEM
1779  452F  DB  D0            10.0          IN      WMCSTA          ;INPUT THE WMC STATUS
1780  4531  E6  80             7.0          ANI     X.DONN          ;XMC-DONE?
1781  4533  CA  3F  45        10.0          JZ      RECCN3          ;YES-EXIT
1782  4536  05                 4.0          DCR     B               ;NO-DECREMENT THE COUNT
1783  4537  C2  2C  45        10.0          JNZ     3$              ;CONTINUE IF NOT ZERO
```

```
1785  453A                                        ERR     T1LOOP,RECCN3
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453A  CD  09  28        18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453D  07                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453E  00                                    .BYTE
 (1)  453F  CD  15  28        18.0        RECCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4542  DA  11  43        10.0                JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1786                                    ;>''XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1787
1788  4545  C3  67  45        10.0                JMP     EXIT
1789
1790  4548                                SEQERR: ERR     T1LOOP,RECCN2
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4548  CD  09  28        18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454B  08                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454C  00                                    .BYTE
 (1)  454D  CD  15  28        18.0        RECCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4550  DA  11  43        10.0                JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1791                                    ;>''XMC NOT DONE'' RESET BEFORE ''WMC NOT DONE'' RESET
1792
1793  4553                                        ENDTST  T1LOOP
 (1)                                              ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4553                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4553  CD  06  28        18.0                CALL    REQST
 (2)  4556  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (2)  4557  00  00                                .WORD                   ;SYSTEM ''*'' COUNT
 (2)  4559  00  00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455B  00                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  455C  07                                    .BYTE   7               ;REQUEST CODE
 (1)  455D  3A  9A  4F        13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  4560  3D                 4.0                DCR     A               ;DOWNCOUNT
 (1)  4561  32  9A  4F        13.0                STA     ITERA           ;SAVE COUNT
 (1)  4564  F2  11  43        10.0                JP      T1LOOP          ;DO TEST UNTIL TILL = 0
1794
1795  4567  C3  18  28        10.0  EXIT:         JMP     TSTEND
1796
1797
1798  456A  3A  87  46        13.0  CLKSYS:       LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
1799  456D  F6  40             7.0                ORI     SSCLK           ;ADD IN 'CLK' BIT
1800  456F  D3  F0            10.0                OUT     CLKCTL          ;LOAD CLOCK CONTROL
1801  4571  E6  BF             7.0                ANI     @277            ;STRIP OFF CLOCK BIT
1802  4573  D3  F0            10.0                OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
1803  4575  C9                10.0                RET                     ;EXIT-CLOCK CYCLE
```

```
1805                                    .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
1806  4576                              SSUB
 (1)                                    ;*********************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
1807                                    ;*CLEAR ALL TU PORTS
1808  4576                              SD
 (1)                                    ;*********************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1809                                    ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
1810                                    ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
1811                                    ;*AND LOOP MODES.
1812  4576                              SP
 (1)                                    ;*********************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1813                                    ;*BGNSUB
1814                                    ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
1815                                    ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
1816                                    ;*     CLEAR PORT SELECT FOR TRANSPORT
1817                                    ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
1818                                    ;*     CLEAR PORT DIAGNOSTIC CONTROL
1819                                    ;*     CLEAR PORT AMTIE WORD
1820                                    ;*ENDSUB
1821  4576                              S
 (1)                                    ;*********************************************************************************
1822  4576  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
1823  4577  C5              12.0          PUSH    B
1824  4578  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
1825
1826  457A  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
1827  457C  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
1828  457E  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1829  457F  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
1830  4581  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
1831  4583  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
1832  4585  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
1833  4586  D3  40          10.0          OUT     TCMD            ;
1834  4588  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
1835  458A  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
1836  458C  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
1837  458E  D3  40          10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
1838  4590  AF               4.0          XRA     A
1839  4591  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
1840  4593  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
1841  4595  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
1842  4597  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
1843  4598  78               4.0          MOV     A,B
1844  4599  FE  04           7.0          CPI     4               ;DONE?
1845  459B  C2  7A  45       10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
1846  459E  C1               10.0          POP     B               ;RESET B & C
1847  459F  F1               10.0          POP     PSW             ;ALL DONE
1848  45A0  C9               10.0          RET                     ;EXIT
```

```
1849
1850
1851
1852  45A1  FF          GXPTBL: .BYTE   $FF          ;PREAMBLE TERM GROUP (10101)
1853                                                 ;CHARACTER 1 = ALL ONES
1854  45A2  FF                  .BYTE   $FF          ;PREAMBLE TERM GROUP
1855                                                 ;CHARACTER 2 = ALL ZEROS
1856  45A3  00                  .BYTE   $00          ;PREAMBLE TERM GROUP
1857                                                 ;CHARACTER 3 = ALL ONES
1858  45A4  00                  .BYTE   $00          ;PREAMBLE TERM GROUP
1859                                                 ;CHARACTER 4 = ALL ZEROS
1860  45A5  FF                  .BYTE   $FF          ;PREAMBLE TERM GROUP
1861                                                 ;CHARACTER 5 = ALL ONES
1862
1863  45A6  FF                  .BYTE   $FF          ;PREAMBLE SECOND GROUP (01111)
1864                                                 ;CHARACTER 1 = ALL ZEROS
1865  45A7  00                  .BYTE   $00          ;PREAMBLE SECOND GROUP
1866                                                 ;CHARACTER 2 = ALL ONES
1867  45A8  FF                  .BYTE   $FF          ;PREAMBLE SECOND GROUP
1868                                                 ;CHARACTER 3 = ALL ONES
1869  45A9  00                  .BYTE   $00          ;PREAMBLE SECOND GROUP
1870                                                 ;CHARACTER 4 = ALL ONES
1871  45AA  FF                  .BYTE   $FF          ;PREAMBLE SECOND GROUP
1872                                                 ;CHARACTER 5 = ALL ONES
1873                                                 ;
1874  45AB  00                  .BYTE   $00          ;PREAMBLE SYNC GROUP 1 (14 GRPS OF 11111)
1875                                                 ;CHARACTER 1 = ALL ONES
1876  45AC  FF                  .BYTE   $FF          ;PREAMBLE SYNC GROUP 1
1877                                                 ;CHARACTER 2 = ALL ONES
1878  45AD  00                  .BYTE   $00          ;PREAMBLE SYNC GROUP 1
1879                                                 ;CHARACTER 3 = ALL ONES
1880  45AE  FF                  .BYTE   $FF          ;PREAMBLE SYNC GROUP 1
1881                                                 ;CHARACTER 4 - ALL ONES
1882  45AF  00                  .BYTE   $00          ;PREAMBLE SYNC GROUP 1
1883                                                 ;CHARACTER 5 = ALL ONES
1884                                                 ;
1885  45B0  FF                  .BYTE   $FF          ;PREAMBLE SYNC GROUP 2
1886                                                 ;CHARACTER 1 = ALL ONES
1887  45B1  00                  .BYTE   $00          ;PREAMBLE SYNC GROUP 2
1888                                                 ;CHARACTER 2 = ALL ONES
1889  45B2  FF                  .BYTE   $FF          ;PREAMBLE SYNC GROUP 2
1890                                                 ;CHARACTER 3 = ALL ONES
1891  45B3  00                  .BYTE   $00          ;PREAMBLE SYNC GROUP 2
1892                                                 ;CHARACTER 4 = ALL ONES
1893  45B4  FF                  .BYTE   $FF          ;PREAMBLE SYNC GROUP 2
1894                                                 ;CHARACTER 5 = ALL ONES
1895                                                 ;
1896  45B5  00                  .BYTE   $00          ;PREAMBLE SYNC GROUP 3
1897                                                 ;CHARACTER 1 = ALL ONES
1898  45B6  FF                  .BYTE   $FF          ;PREAMBLE SYNC GROUP 3
1899                                                 ;CHARACTER 2 = ALL ONES
1900  45B7  00                  .BYTE   $00          ;PREAMBLE SYNC GROUP 3
1901                                                 ;CHARACTER 3 = ALL ONES
1902  45B8  FF                  .BYTE   $FF          ;PREAMBLE SYNC GROUP 3
```

```
1903
1904  45B9  00          .BYTE   $00     ;CHARACTER 4 - ALL ONES
1905                                    ;PREAMBLE SYNC GROUP 3
1906                                    ;CHARACTER 5 = ALL ONES
1907  45BA  FF          .BYTE   $FF     ;
1908                                    ;PREAMBLE SYNC GROUP 4
1909  45BB  00          .BYTE   $00     ;CHARACTER 1 = ALL ONES
1910                                    ;PREAMBLE SYNC GROUP 4
1911  45BC  FF          .BYTE   $FF     ;CHARACTER 2 = ALL ONES
1912                                    ;PREAMBLE SYNC GROUP 4
1913  45BD  00          .BYTE   $00     ;CHARACTER 3 = ALL ONES
1914                                    ;PREAMBLE SYNC GROUP 4
1915  45BE  FF          .BYTE   $FF     ;CHARACTER 4 = ALL ONES
1916                                    ;PREAMBLE SYNC GROUP 4
1917                                    ;CHARACTER 5 = ALL ONES
1918  45BF  00          .BYTE   $00     ;
1919                                    ;PREAMBLE SYNC GROUP 5
1920  45C0  FF          .BYTE   $FF     ;CHARACTER 1 = ALL ONES
1921                                    ;PREAMBLE SYNC GROUP 5
1922  45C1  00          .BYTE   $00     ;CHARACTER 2 = ALL ONES
1923                                    ;PREAMBLE SYNC GROUP 5
1924  45C2  FF          .BYTE   $FF     ;CHARACTER 3 = ALL ONES
1925                                    ;PREAMBLE SYNC GROUP 5
1926  45C3  00          .BYTE   $00     ;CHARACTER 4 - ALL ONES
1927                                    ;PREAMBLE SYNC GROUP 5
1928                                    ;CHARACTER 5 = ALL ONES
1929  45C4  FF          .BYTE   $FF     ;
1930                                    ;PREAMBLE SYNC GROUP 6
1931  45C5  00          .BYTE   $00     ;CHARACTER 1 = ALL ONES
1932                                    ;PREAMBLE SYNC GROUP 6
1933  45C6  FF          .BYTE   $FF     ;CHARACTER 2 = ALL ONES
1934                                    ;PREAMBLE SYNC GROUP 6
1935  45C7  00          .BYTE   $00     ;CHARACTER 3 = ALL ONES
1936                                    ;PREAMBLE SYNC GROUP 6
1937  45C8  FF          .BYTE   $FF     ;CHARACTER 4 = ALL ONES
1938                                    ;PREAMBLE SYNC GROUP 6
1939                                    ;CHARACTER 5 = ALL ONES
1940  45C9  00          .BYTE   $00     ;
1941                                    ;PREAMBLE SYNC GROUP 7
1942  45CA  FF          .BYTE   $FF     ;CHARACTER 1 = ALL ONES
1943                                    ;PREAMBLE SYNC GROUP 7
1944  45CB  00          .BYTE   $00     ;CHARACTER 2 = ALL ONES
1945                                    ;PREAMBLE SYNC GROUP 7
1946  45CC  FF          .BYTE   $FF     ;CHARACTER 3 = ALL ONES
1947                                    ;PREAMBLE SYNC GROUP 7
1948  45CD  00          .BYTE   $00     ;CHARACTER 4 - ALL ONES
1949                                    ;PREAMBLE SYNC GROUP 7
1950                                    ;CHARACTER 5 = ALL ONES
1951  45CE  FF          .BYTE   $FF     ;
1952                                    ;PREAMBLE SYNC GROUP 8
1953  45CF  00          .BYTE   $00     ;CHARACTER 1 = ALL ONES
1954                                    ;PREAMBLE SYNC GROUP 8
1955  45D0  FF          .BYTE   $FF     ;CHARACTER 2 = ALL ONES
1956                                    ;PREAMBLE SYNC GROUP 8
                                        ;CHARACTER 3 = ALL ONES
```

```
1957  45D1  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 8
1958                                        ;CHARACTER 4 = ALL ONES
1959  45D2  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 8
1960                                        ;CHARACTER 5 = ALL ONES
1961                                        ;
1962  45D3  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 9
1963                                        ;CHARACTER 1 = ALL ONES
1964  45D4  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 9
1965                                        ;CHARACTER 2 = ALL ONES
1966  45D5  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 9
1967                                        ;CHARACTER 3 = ALL ONES
1968  45D6  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 9
1969                                        ;CHARACTER 4 = ALL ONES
1970  45D7  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 9
1971                                        ;CHARACTER 5 = ALL ONES
1972                                        ;
1973  45D8  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 10
1974                                        ;CHARACTER 1 = ALL ONES
1975  45D9  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 10
1976                                        ;CHARACTER 2 = ALL ONES
1977  45DA  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 10
1978                                        ;CHARACTER 3 = ALL ONES
1979  45DB  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 10
1980                                        ;CHARACTER 4 = ALL ONES
1981  45DC  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 10
1982                                        ;CHARACTER 5 = ALL ONES
1983                                        ;
1984  45DD  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 11
1985                                        ;CHARACTER 1 = ALL ONES
1986  45DE  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 11
1987                                        ;CHARACTER 2 = ALL ONES
1988  45DF  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 11
1989                                        ;CHARACTER 3 = ALL ONES
1990  45E0  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 11
1991                                        ;CHARACTER 4 = ALL ONES
1992  45E1  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 11
1993                                        ;CHARACTER 5 = ALL ONES
1994                                        ;
1995  45E2  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 12
1996                                        ;CHARACTER 1 = ALL ONES
1997  45E3  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 12
1998                                        ;CHARACTER 2 = ALL ONES
1999  45E4  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 12
2000                                        ;CHARACTER 3 = ALL ONES
2001  45E5  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 12
2002                                        ;CHARACTER 4 = ALL ONES
2003  45E6  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 12
2004                                        ;CHARACTER 5 = ALL ONES
2005                                        ;
2006  45E7  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 13
2007                                        ;CHARACTER 1 = ALL ONES
2008  45E8  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 13
2009                                        ;CHARACTER 2 = ALL ONES
2010  45E9  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 13
```

```
2011
2012  45EA  FF                        .BYTE   $FF            ;CHARACTER 3 = ALL ONES
2013                                                         ;PREAMBLE SYNC GROUP 13
2014  45EB  00                        .BYTE   $00            ;CHARACTER 4 - ALL ONES
2015                                                         ;PREAMBLE SYNC GROUP 13
2016                                                         ;CHARACTER 5 = ALL ONES
2017  45EC  FF                        .BYTE   $FF            ;
2018                                                         ;PREAMBLE SYNC GROUP 14
2019  45ED  00                        .BYTE   $00            ;CHARACTER 1 = ALL ONES
2020                                                         ;PREAMBLE SYNC GROUP 14
2021  45EE  FF                        .BYTE   $FF            ;CHARACTER 2 = ALL ONES
2022                                                         ;PREAMBLE SYNC GROUP 14
2023  45EF  00                        .BYTE   $00            ;CHARACTER 3 = ALL ONES
2024                                                         ;PREAMBLE SYNC GROUP 14
2025  45F0  FF                        .BYTE   $FF            ;CHARACTER 4 = ALL ONES
2026                                                         ;PREAMBLE SYNC GROUP 14
2027                                                         ;CHARACTER 5 = ALL ONES
2028  45F1  FF                        .BYTE   $FF            ;
2029                                                         ;MARK ONE GROUP
2030  45F2  FF                        .BYTE   $FF            ;CHARACTER 1 = ALL ZEROS
2031                                                         ;MARK ONE GROUP
2032  45F3  00                        .BYTE   $00            ;CHARACTER 2 = ALL ZEROS
2033                                                         ;MARK ONE GROUP
2034  45F4  FF                        .BYTE   $FF            ;CHARACTER 3 = ALL ONES
2035                                                         ;MARK ONE GROUP
2036  45F5  00                        .BYTE   $00            ;CHARACTER 4 = ALL ONES
2037                                                         ;MARK ONE GROUP
2038                                                         ;CHARACTER 5 = ALL ONES
2039                                ;THERE IS NO DATA GROUP ... ONLY 1 DATA BYTE IN THE RESIDUAL GROUP.
2040                                ;ECC FOR DATA GRP = 0.
2041                                ;THIS IS TRANSITION # 86.
2042
2043                                ;END MARK = 11111 (ALL 1'S)
2044
2045  45F6  FF                              .BYTE   $FF      ;END MARK GROUP
2046                                                         ;CHARACTER 1 = ALL ONES
2047  45F7  00                              .BYTE   $00      ;END MARK GROUP
2048                                                         ;CHARACTER 2 = ALL ONES
2049  45F8  FF                              .BYTE   $FF      ;END MARK GROPU
2050                                                         ;CHARACTER 3 = ALL ONES
2051  45F9  00                              .BYTE   $00      ;END MARK GROUP
2052                                                         ;CHARACTER 4 = ALL ONES
2053  45FA  FF                              .BYTE   $FF      ;END MARK GROUP
2054                                                         ;CHARACTER 5 = ALL ONES
2055                                                         ;
2056
2057
2058                                ;1 DATA BYTE AND 3 PAD BYTES OF ALL 0'S (11001)
2059                                ;TRANSITION #91.
2060
2061  45FB  00                      GXPDAT: .BYTE   @0       ;RESIDUAL GROUP - AFTER TRANSLATION - 1ST HALF
2062                                                         ;CHARACTER 1
2063  45FC  FF                              .BYTE   $FF      ;RESIDUAL GROUP - AFTER TRANSLATION
2064                                                         ;CHARACTER 2
```

```
2065  45FD  FF                        .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2066                                                          ;CHARACTER 3
2067  45FE  FF                        .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2068                                                          ;CHARACTER 4
2069  45FF  00                        .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION
2070                                                          ;CHARACTER 5
2071
2072                          ;2 PAD BYTES
2073                          ;ACRC = XXX
2074                          ;ECC = XXX
2075                          ;TRANSITION #96.
2076
2077  4600  FF                        .BYTE   @377            ;RESIDUAL GROUP - AFTER TRANSLATION - 2ND HALF
2078                                                          ;CHARACTER 1
2079  4601  BD                        .BYTE   @275            ;RESIDUAL GROUP - AFTER TRANSLATION
2080                                                          ;CHARACTER 2
2081  4602  BD                        .BYTE   @275            ;RESIDUAL GROUP - AFTER TRANSLATION
2082                                                          ;CHARACTER 3
2083  4603  42                        .BYTE   @102            ;RESIDUAL GROUP - AFTER TRANSLATION
2084                                                          ;CHARACTER 4
2085  4604  A4                        .BYTE   @244            ;RESIDUAL GROUP - AFTER TRANSLATION
2086                                                          ;CHARACTER 5
2087                                                          ;
2088                          ;4 CRC BYTES OF XXX
2089                          ;TRANSITION #101.
2090
2091  4605  10                        .BYTE   @020            ;CRC GROUP - AFTER TRANSLATION - 1ST HALF
2092                                                          ;CHARACTER 1
2093  4606  EF                        .BYTE   @357            ;CRC GROUP - AFTER TRANSLATION
2094                                                          ;CHARACTER 2
2095  4607  A4                        .BYTE   @244            ;CRC GROUP - AFTER TRANSLATION
2096                                                          ;CHARACTER 3
2097  4608  EF                        .BYTE   @357            ;CRC GROUP - AFTER TRANSLATION
2098                                                          ;CHARACTER 4
2099  4609  10                        .BYTE   @020            ;CRC GROUP - AFTER TRANSLATION
2100                                                          ;CHARACTER 5
2101
2102                          ;2 CRC OF XXX
2103                          ;RESID OF 000
2104                          ;ECC OF XXX
2105                          ;TRANSITION #106.
2106
2107  460A  E6                        .BYTE   @346            ;CRC GROUP - AFTER TRANSLATION - 2ND HALF
2108                                                          ;CHARACTER 6
2109  460B  39                        .BYTE   @071            ;CRC GROUP - AFTER TRANSLATION
2110                                                          ;CHARACTER 7
2111  460C  72                        .BYTE   @162            ;CRC GROUP - AFTER TRANSLATION
2112                                                          ;CHARACTER 8
2113  460D  14                        .BYTE   @024            ;CRC GROUP - AFTER TRANSLATION
2114                                                          ;CHARACTER 9
2115  460E  A9                        .BYTE   @251            ;CRC GROUP - AFTER TRANSLATION
2116                                                          ;CHARACTER 10
2117                                                          ;
2118
```

```
2119
2120  460F  56                          .BYTE   @126        ;MARK TWO
                                                           ;CHARACTER 1 = ALL ONES
2121
2122  4610  A9                          .BYTE   @251        ;MARK TWO
                                                           ;CHARACTER 2 = ALL ONES
2123
2124  4611  56                          .BYTE   @126        ;MARK TWO
                                                           ;CHARACTER 3 = ALL ONES
2125
2126  4612  56                          .BYTE   @126        ;MARK TWO
                                                           ;CHARACTER 4 = ALL ZEROS
2127
2128  4613  56                          .BYTE   @126        ;MARK TWO
                                                           ;CHARACTER 5 = ALL ZEROS
2129
2130                                                       ;
2131  4614  A9                    GDATEN: .BYTE   @251        ;POSTAMBLE SYNC GROUP 1
                                                           ;CHARACTER 1 = ALL ONES
2132
2133  4615  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 1
                                                           ;CHARACTER 2 - ALL ONES
2134
2135  4616  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 1
                                                           ;CHARACTER 3 = ALL ONES
2136
2137  4617  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 1
                                                           ;CHARACTER 4 = ALL ONES
2138
2139  4618  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 1
                                                           ;CHARACTER 5 = ALL ONES
2140
2141                                                       ;
2142  4619  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 2
                                                           ;CHARACTER 1 = ALL ONES
2143
2144  461A  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 2
                                                           ;CHARACTER 2 - ALL ONES
2145
2146  461B  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 2
                                                           ;CHARACTER 3 = ALL ONES
2147
2148  461C  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 2
                                                           ;CHARACTER 4 = ALL ONES
2149
2150  461D  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 2
                                                           ;CHARACTER 5 = ALL ONES
2151
2152                                                       ;
2153  461E  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 3
                                                           ;CHARACTER 1 = ALL ONES
2154
2155  461F  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 3
                                                           ;CHARACTER 2 - ALL ONES
2156
2157  4620  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 3
                                                           ;CHARACTER 3 = ALL ONES
2158
2159  4621  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 3
                                                           ;CHARACTER 4 = ALL ONES
2160
2161  4622  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 3
                                                           ;CHARACTER 5 = ALL ONES
2162
2163                                                       ;
2164  4623  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 4
                                                           ;CHARACTER 1 = ALL ONES
2165
2166  4624  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 4
                                                           ;CHARACTER 2 - ALL ONES
2167
2168  4625  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 4
                                                           ;CHARACTER 3 = ALL ONES
2169
2170  4626  A9                          .BYTE   @251        ;POSTAMBLE SYNC GROUP 4
                                                           ;CHARACTER 4 = ALL ONES
2171
2172  4627  56                          .BYTE   @126        ;POSTAMBLE SYNC GROUP 4
```

```
2173                                                    ;CHARACTER 5 = ALL ONES
2174                                                    ;
2175  4628  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 5
2176                                                    ;CHARACTER 1 = ALL ONES
2177  4629  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 5
2178                                                    ;CHARACTER 2 - ALL ONES
2179  462A  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 5
2180                                                    ;CHARACTER 3 = ALL ONES
2181  462B  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 5
2182                                                    ;CHARACTER 4 = ALL ONES
2183  462C  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 5
2184                                                    ;CHARACTER 5 = ALL ONES
2185                                                    ;
2186  462D  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 6
2187                                                    ;CHARACTER 1 = ALL ONES
2188  462E  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 6
2189                                                    ;CHARACTER 2 - ALL ONES
2190  462F  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 6
2191                                                    ;CHARACTER 3 = ALL ONES
2192  4630  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 6
2193                                                    ;CHARACTER 4 = ALL ONES
2194  4631  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 6
2195                                                    ;CHARACTER 5 = ALL ONES
2196                                                    ;
2197  4632  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 7
2198                                                    ;CHARACTER 1 = ALL ONES
2199  4633  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 7
2200                                                    ;CHARACTER 2 - ALL ONES
2201  4634  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 7
2202                                                    ;CHARACTER 3 = ALL ONES
2203  4635  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 7
2204                                                    ;CHARACTER 4 = ALL ONES
2205  4636  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 7
2206                                                    ;CHARACTER 5 = ALL ONES
2207                                                    ;
2208  4637  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 8
2209                                                    ;CHARACTER 1 = ALL ONES
2210  4638  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 8
2211                                                    ;CHARACTER 2 - ALL ONES
2212  4639  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 8
2213                                                    ;CHARACTER 3 = ALL ONES
2214  463A  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 8
2215                                                    ;CHARACTER 4 = ALL ONES
2216  463B  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 8
2217                                                    ;CHARACTER 5 = ALL ONES
2218                                                    ;
2219  463C  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 9
2220                                                    ;CHARACTER 1 = ALL ONES
2221  463D  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 9
2222                                                    ;CHARACTER 2 - ALL ONES
2223  463E  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 9
2224                                                    ;CHARACTER 3 = ALL ONES
2225  463F  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 9
2226                                                    ;CHARACTER 4 = ALL ONES
```

```
2227  4640  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 9
2228                                        ;CHARACTER 5 = ALL ONES
2229                                        ;
2230  4641  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 10
2231                                        ;CHARACTER 1 = ALL ONES
2232  4642  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 10
2233                                        ;CHARACTER 2 - ALL ONES
2234  4643  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 10
2235                                        ;CHARACTER 3 = ALL ONES
2236  4644  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 10
2237                                        ;CHARACTER 4 = ALL ONES
2238  4645  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 10
2239                                        ;CHARACTER 5 = ALL ONES
2240                                        ;
2241  4646  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 11
2242                                        ;CHARACTER 1 = ALL ONES
2243  4647  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 11
2244                                        ;CHARACTER 2 - ALL ONES
2245  4648  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 11
2246                                        ;CHARACTER 3 = ALL ONES
2247  4649  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 11
2248                                        ;CHARACTER 4 = ALL ONES
2249  464A  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 11
2250                                        ;CHARACTER 5 = ALL ONES
2251                                        ;
2252  464B  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 12
2253                                        ;CHARACTER 1 = ALL ONES
2254  464C  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 12
2255                                        ;CHARACTER 2 - ALL ONES
2256  464D  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 12
2257                                        ;CHARACTER 3 = ALL ONES
2258  464E  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 12
2259                                        ;CHARACTER 4 = ALL ONES
2260  464F  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 12
2261                                        ;CHARACTER 5 = ALL ONES
2262                                        ;
2263  4650  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 13
2264                                        ;CHARACTER 1 = ALL ONES
2265  4651  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 13
2266                                        ;CHARACTER 2 - ALL ONES
2267  4652  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 13
2268                                        ;CHARACTER 3 = ALL ONES
2269  4653  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 13
2270                                        ;CHARACTER 4 = ALL ONES
2271  4654  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 13
2272                                        ;CHARACTER 5 = ALL ONES
2273                                        ;
2274  4655  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 14
2275                                        ;CHARACTER 1 = ALL ONES
2276  4656  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 14
2277                                        ;CHARACTER 2 - ALL ONES
2278  4657  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 14
2279                                        ;CHARACTER 3 = ALL ONES
2280  4658  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 14
```

```
2281
2282  4659  56                          .BYTE   @126    ;CHARACTER 4 = ALL ONES
2283                                                    ;POSTAMBLE SYNC GROUP 14
2284                                                    ;CHARACTER 5 = ALL ONES
2285  465A  A9                          .BYTE   @251    ;
2286                                                    ;POSTAMBLE SECOND GROUP
2287  465B  56                          .BYTE   @126    ;CHARACTER 1 = ALL ONES
2288                                                    ;POSTAMBLE SECOND GROUP
2289  465C  A9                          .BYTE   @251    ;CHARACTER 2 = ALL ONES
2290                                                    ;POSTAMBLE SECOND GROUP
2291  465D  56                          .BYTE   @126    ;CHARACTER 3 = ALL ONES
2292                                                    ;POSTAMBLE SECOND GROUP
2293  465E  56                          .BYTE   @126    ;CHARACTER 4 = ALL ONES
2294                                                    ;POSTAMBLE SECOND GROUP
2295                                                    ;CHARACTER 5 = ALL ZEROS
2296  465F  A9                          .BYTE   @251    ;
2297                                                    ;POSTAMBLE TERM GROUP
2298  4660  A9                          .BYTE   @251    ;CHARACTER 1 = ALL ONES
2299                                                    ;POSTAMBLE TERM GROUP
2300  4661  56                          .BYTE   @126    ;CHARACTER 2 = ALL ZEROS
2301                                                    ;POSTAMBLE TERM GROUP
2302  4662  56                          .BYTE   @126    ;CHARACTER 3 = ALL ONES
2303                                                    ;POSTAMBLE TERM GROUP
2304  4663  56              GTBLEN:     .BYTE   @126    ;CHARACTER 4 = ALL ZEROS
2305                                                    ;POSTAMBLE TERM GROUP
                                                        ;CHARACTER 5 = 0'S TO RESET TRACKS
```

```
                                        .SBTTL  TABLE EXPECTED PARITY

                                        ;HERE IS THE TABLE OF EXPECTED PARITY STATES FOR RESIDUAL GROUP
                                        ;NO BYTES IN DATA GROUP.

4664  00                                PXPTBL: .BYTE   0               ;THIS IS A DUMMY ENTRY

                                        ;RESIDUAL GRP
                                        ;1ST GRP IS 1 DATA BYTE OF 0 AND 3 PADS SO 4 X 5 IS 0000 = 11001 FOR DATA
                                        ;AND PARITY IS 1111 = 01111

4665  01                                        .BYTE   1
4666  00                                        .BYTE   0
4667  01                                        .BYTE   1
4668  00                                        .BYTE   0
4669  01                                        .BYTE   1

                                        ;2ND GRP IS 2 PAD BYTES + ACRC OF XXX + ECC OF XXX
                                        ;'P' 4 X 5 IS

466A  01                                        .BYTE   1
466B  00                                        .BYTE   0
466C  01                                        .BYTE   1
466D  00                                        .BYTE   0
466E  00                                        .BYTE   0

                                        ;CRC GRP 1ST HALF IS 4 CRC BYTES OF XXX
                                        ;'P' 4 X 5 IS

466F  00                                        .BYTE   0
4670  01                                        .BYTE   1
4671  00                                        .BYTE   0
4672  01                                        .BYTE   1
4673  00                                        .BYTE   0

                                        ;2ND HALF OF CRC GRP IS 2 CRC OF XXX + RESID BYTE OF XXX + ECC BYTE OF XXX
                                        ;'P' 4 X 5 IS

4674  00                                        .BYTE   0
4675  01                                        .BYTE   1
4676  00                                        .BYTE   0
4677  00                                        .BYTE   0
4678  01                                        .BYTE   1

                                        ;HERE IS THE MARK-2 PARITY TABLE

4679  00                                        .BYTE   0               ;MARK 2 IS 11100
467A  01                                        .BYTE   1
467B  00                                        .BYTE   0
467C  00                                        .BYTE   0
467D  00                                        .BYTE   0
```

```
                                        .SBTTL  PROGRAM VARIABLES

                                UNITMP: .BYTE   0               ;UNIT MAP
                                XDATA:  .BYTE   0               ;COPY OF EXPECTED DATA
                                DPCNT:  .WORD   0               ;PARITY TEST COUNTER
                                EXDLPC: .BYTE   0               ;TRANSITION COUNT AT ERROR
                                POINTER: .WORD  0               ;POINTER TO EXPECTED DATA TABLE
                                TRANSCNT: .WORD 0               ;TRANSITION COUNT
                                CCTLWD: .BYTE   0               ;CLOCK CONTROL WORD

                                        .END
```

2360				
2361				
2362	467E	00		
2363	467F	00		
2364	4680	00	00	
2365	4682	00		
2366	4683	00	00	
2367	4685	00	00	
2368	4687	00		
2369				
2370		0000		

```
A      =%0007          ABOCN0  43E2  G        ADATA  = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098           ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000          BADST  = 0090          BGNPRE  43EB
BIT0   = 0001          BIT1   = 0002          BIT15  = 8000          BIT2   = 0004
BIT3   = 0008          BIT4   = 0010          BIT5   = 0020          BIT6   = 0040
BIT7   = 0080          BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A
BRKRAM= 4F10           BRKSTR= 4E60           BRKXCT= 4F00           BSAVE   4F9C
BYTCNT= 00D4           BYTEH   4F24           BYTEL   4F23           C      =%0001
CASCT   4F21           CASCTL= 00A0           CASSTA= 00A0           CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1           CBYTH  = 008B          CBYTL  = 008A
CCTLWD  4687           CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028           CH0TIE= 0020           CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023           CH4TIE= 0024           CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027           CKLOP  = 2815          CLEAR   4576
CLK     43CF           CLKCTL= 00F0           CLKRS   43C2          CLKSYS  456A
CLK1    43CD           CLOCK   4F26           CLRLP   457A          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7           CRCWRD= 0018           CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082          C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040
C.WCS  = 0002          D      =%0002          DATACT= 00D0           DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB           DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 0008          DONE1  = 0045          DONINT= 0010
DPCNT   4680           DPTST   4476           DSAVE   4F9E          DSE    = 0006
DUMMY   4367  G        D.ATH0= 0001           D.ATH1= 0002           D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004           D.TACH= 0008
D.WR4  = 0080          E      =%0003          ECCBAD= 0042           ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E           EDATA  = 0095
EOTCLR= 0003           ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90          ERRCNT= 00D6
ESAVE   4F9F           EXDLPC  4682           EXIT    4567          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001           E.TTEC= 0002           E.UNC  = 0004
FIFORD= 006A           FORMAT  4F25           FOUND   4370          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002           GDATEN  4614          GOODTM= 0092
GTBLEN  4663           GXPDAT  45FB           GXPTBL  45A1          H      =%0004
HLSAVE  4FA0           IE     = 0008          INTSTA= 00E0           ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020
I7.5   = 0040          KCALL  = 005F          KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8           KEY1   = 0078
KEY10  = 006D          KEY11  = 006E          KEY12  = 006F          KEY13  = 005C
KEY14  = 005D          KEY15  = 005E          KEY16  = 005F          KEY17  = 003C
KEY18  = 003D          KEY19  = 003E          KEY2   = 0079          KEY20  = 003F
KEY3   = 007A          KEY4   = 007B          KEY5   = 0074          KEY6   = 0075
KEY7   = 0076          KEY8   = 0077          KEY9   = 006C          KINTA  = 006F
```

```
KLDAD = 003D            KN0   = 003C            KN1   = 005C            KN2   = 005D
KN3   = 005E            KN4   = 006C            KN5   = 006D            KN6   = 006E
KN7   = 0074            KN8   = 0075            KN9   = 0076            KU2   = 0079
KU3   = 007A            KU8   = 0077            L     =%0005            LBLANK= 000F
LCE   = 0008            LCH   = 000C            LCL   = 000D            LCP   = 000E
LC0   = 0000            LC1   = 0001            LC2   = 0002            LC3   = 0003
LC4   = 0004            LC5   = 0005            LC6   = 0006            LC7   = 0007
LC8   = 0008            LC9   = 0009            LDLEDA= 00CA            LDLEDB= 00CB
LDLEDC= 00CC            LDLEDD= 00CD            LDLEDE= 00CE            LDLEDF= 00CF
LKDIAG= 2800            LKKBD = 004C            LKKEY = 0049            LKLWMG= 0058
LKLWMP= 0055            LKLWPG= 0052            LKLWPP= 004F            LKMOD7= 0046
LKOPR = 0046            LPFLG   4F94            LPNUM   4F92            M     =%0006
MBSEL = 00E0            MB.A  = 0008            MB.B  = 0004            MEMTOP= 4FFF
MINUS = 000A            MM    = 8000            MSE   = 0008            MSGN  = 0008
MTACLR= 0000            MT.ARA= 0020            MT.CPE= 0080            MT.DSE= 0001
MT.FWD= 0040            MT.INH= 0008            MT.LWR= 0004            MT.MOT= 0002
MT.NWT= 0080            MT.PEC= 0040            MT.PSB= 0004            MT.PS0= 0001
MT.PS1= 0002            MT.REV= 0020            MT.WRT= 0010            MT.Z  = 0008
M.ATA = 0080            M.CAPE= 0020            M.CONT= 0080            M.DEM = 0020
M.EBL = 0010            M.EXC = 0008            M.FAIL= 0008            M.ILR = 0010
M.INIT= 0010            M.OCC = 0020            M.ONLI= 0001            M.PE  = 0040
M.PORT= 0080            M.RDEN= 0002            M.RDPE= 0008            M.RUN = 0004
M.SCLK= 0001            M.TRA = 0040            M.UNIT= 0007            M.WCLK= 0040
M.WCLN= 0080            M.WREN= 0080            M5.5  = 0001            M6.5  = 0002
M7.5  = 0004            NOTCAP= 0088            OKAY  = 00FF            OPRRAM= 4300
OPSTRT= 0058            OPVER = 0040            PADCNT= 00D5            PADCRC= 0080
PDIAG = 0048            PEID  = 008A            PENAB = 004C            PESET = 0001
PL    = 00B1            POINTE  4683            PPOST   4489            PPREAM  4489
PRDD  = 004C            PRECN1  4426  G         PRECN3  44EE  G         PRENF = 009C
PS    = 00B2            PSTAT = 0048            PSW   =%0009            PTSTOK  44AE  G
PTY     4491            PTYC    44D2            PTY1    449C            PTY2    44A9
PXPTBL  4664            P.AMTP= 0001            P.BCTC= 0040            P.CMDP= 0020
P.INTE= 0080            P.LCS = 0040            P.LWR = 0020            P.RDP = 0002
P.RPEN= 0004            P.RPST= 0002            P.RPOE= 0020            P.RP1E= 0010
P.RP2E= 0020            P.RP3E= 0010            P.SING= 0080            P.STAT= 0002
P.STPE= 0080            P.TACH= 0008            P.TUPR= 0010            P.WCSP= 0004
P.WDS = 0040            P.WFLP= 0001            P.WPEN= 0010            P.WPOE= 0008
P.WP1E= 0004            P.WP2E= 0008            P.WP3E= 0004            P.5VOK= 0002
QUE   = 281B            QUEM  = 281E            RAMT  = 0010            RARA  = 0006
RARAI = 0004            RCHBD0= 0048            RCHBD1= 0047            RCHOK = 0046
RCHTST= 000C            RCLRT = 000D            RCMD  = 000B            RCMLP = 0003
RCONT = 0080            RDATA = 0017            RDCLK = 0010            RDON  = 0011
READG = 0007            RECCN1  4524  G         RECCN2  454D  G         RECCN3  453F  G
RECEND  4508            REND  = 0014            REQST = 2806            RESCHR= 00D1
REVTST= 0064            REWIND= 0004            RFIFOL= 0008            RGCLK = 0002
RGCRI = 0003            RIBG  = 0001            RILL  = 0012            RINST = 000C
RMCTST= 0008            RMK2  = 0013            RNOP  = 0000            RPATH = 0001
RPBAD = 0044            RPCHI = 0001            RPCLK = 0003            RPCTL = 0009
RPEI  = 0002            RPFAIL= 0000            RPF1  = 009D            RPF2  = 009E
RPOK  = 0043            RPOSTN= 0016            RPSTA = 0015            RRCMT = 000A
RSTAT = 0002            RTIEB = 000A            RTIER = 0030            RTM   = 0005
RUNKI = 0009            RUPTST= 005E            RWDUNL= 0005            R.AMT = 0001
R.BOP = 0008            R.DATA= 0040            R.DON = 0002            R.DRDY= 0010
R.END = 0010            R.ILL = 0004            R.JVOK= 0004            R.MK2 = 0008
```

XMC4 - TRANSLATOR MICRO CONTROLLER PART #4 CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:10 PAGE 2-4 SEQ 0511
XMC4.M80 SYMBOL TABLE

```
R.PLOD= 0008    R.PL00= 0010    R.PL01= 0020    R.POST= 0020
R.STNM= 0002    R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080
R.TSTD= 0040    R.VOK = 0080    R00H  = 0081    R00L  = 0080
R01H  = 0083    R01L  = 0082    R02H  = 0085    R02L  = 0084
R03H  = 0087    R03L  = 0086    R04H  = 0089    R04L  = 0088
R05H  = 008B    R05L  = 008A    R06H  = 008D    R06L  = 008C
R07H  = 008F    R07L  = 008E    R10H  = 0091    R10L  = 0090
R11H  = 0093    R11L  = 0092    R12H  = 0095    R12L  = 0094
R13H  = 0097    R13L  = 0096    R14H  = 0099    R14L  = 0098
R15H  = 009B    R15L  = 009A    R16H  = 009D    R16L  = 009C
R17H  = 009F    R17L  = 009E    R7.5  = 0010    SELCLR= 0000
SEQERR  4548    SETATA= 00A1    SID   = 0080    SOD   = 0080
SOE   = 0040    SP    =%0008    SSCLK = 0040    SSTEP = 0005
STACK = 4FFF    STATRM= 4F20    STPCT   4F20    STRSP = 5000
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TMF   = 0099    TMRDY = 0040
TRANSC  4685    TRKENA= 00D2    TSET  = 2803    TSTEND= 2818
TSTS  = 0040    TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010
T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002
T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020    T.PES = 0008
T.PSBJ- 0020    T.PS0J= 0008    T.PS1J= 0010    T.RDY = 0080
T.RDY0= 0040    T.RWD = 0010    T.SCLK= 0002    T1LOOP  4311
T1LPA   433F    UIBG  = 00A1    UNITMP  467E    VALFC   4F98
VALTB   4F95    VELTST= 005B    WDR.P = 0010    WMCCTL= 00D3
WMCERR= 00DA    WMCSTA= 00D0    WRTCLK= 0000    WRTDAT= 00D3
W.ACRC= 0004    W.CRC = 0008    W.DIAG= 0002    W.DONN= 0040
W.ECC = 0010    W.ENAB= 0080    W.ERR = 0020    W.FMT = 0070
W.GCR = 0010    W.LEFT= 0004    W.ONES= 0020    W.RESI= 0002
W.REV = 0004    W.ROME= 0010    W.RST = 0001    W.SKIP= 000F
W.WRIT= 0008    W.XFER= 0020    X     =%000A    XDATA   467F
XDLOOP  442F    XDLPC   444A    XDLPC1  4414    XDLPC2  4421
XDLP1   443C    XPLOOP  44B7    XPLP1   44C4    X.DONN= 0080
X.ENAB= 0040    X.PEPE= 0002    X.ROME= 0001    X.WCLK= 0001
Y     =%0008    .     = 4688
```

ERRORS DETECTED: 0

*XMC4.A78/PTP,XMC4=NLIST,PARAM,MACRO,LIST,XMC4
RUN-TIME: 4 5 0 SECONDS
CORE USED: 10K

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:10 SEQ 0512
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  XMC5 - WRITE PATH 4 X 5 TRANSLATIONS
                                .MACRO  SETLPC  ARG1,ARG2
                                        LXI     H,ARG1          ;GET THE LOOP-ON-ERR ADDRESS
                                        SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
                                        LXI     H,ARG2          ;GET THE CONTINUE ADDRESS
                                        SHLD    ERCADR+1        ;SAVE IT
                                        XRA     A
                                .ENDM
                                .SBTTL  TEST 1 - XMC - GCR 4X5 TRANSLATION SUBGROUP 0000(2) TEST
                                ;ID     XMC5-TRANSLATOR MICRO CONTROLLER PART #5
1339  4300                      ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
1341  4300                      SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0000(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 11001(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
1352  4300                      SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
```

```
1374                    ;*      :       :       DECREMENT THE LOOP COUNT
1375                    ;*      :       :       DO UNTIL LOOP COUNTER = 0
1376                    ;*      :       :ENDDO
1377                    ;*      :       :ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
1378                    ;*      :       :BGNDO
1379                    ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
1380                    ;*      :       :       BGNDO
1381                    ;*      :       :       :       CLOCK THE SYSTEM
1382                    ;*      :       :       :       DECREMENT THE LOOP COUNT
1383                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1384                    ;*      :       :       ENDDO
1385                    ;*      :       :       IF TIMEOUT COUNT = 0
1386                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1387                    ;*      :       :       :       ELSE - CONTINUE
1388                    ;*      :       :       ENDIF
1389                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1390                    ;*      :       :ENDDO
1391                    ;*      :       :BGNDO
1392                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1393                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1394                    ;*      :       :       :       THEN - CONTINUE
1395                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1396                    ;*      :       :       ENDIF
1397                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1398                    ;*      :       :       BGNDO
1399                    ;*      :       :       :       CLOCK THE SYSTEM
1400                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1401                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1402                    ;*      :       :       ENDDO
1403                    ;*      :       :       IF 'WCLK' IS SET
1404                    ;*      :       :       :       THEN - CONTINUE
1405                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1406                    ;*      :       :       ENDIF
1407                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1408                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1409                    ;*      :       :ENDDO
1410                    ;*      :       :POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1411                    ;*      :       :DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1412                    ;*      :ENDDO
1413                    ;*ENDTST
1414  4300              SE
 (1)                    ;*****************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1415                    ;*XMC5 MICRO TEST 01
1416                    ;*XMC5 MICRO ERROR 01
1417                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
1418                    ;*M8958, M8959, M8955'S
1419                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1420                    ;*ACTUAL = NNNN
1421                    ;*EXPECTED = NNNN
1422                    ;*TRANSITION COUNT = LLL
1423                    ;*
1424                    ;*XMC5 MICRO TEST 01
```

```
                                        ;*XMC5 MICRO ERROR 21
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*XMC5 MICRO TEST 01
                                        ;*XMC5 MICRO ERROR 22
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 01
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
                                        S
1443  4300                              ; ***************************************************************************
 (1)                                    TEST1:  TESTX   1
 (1)  4300  3E  01          7.0                          MVI     A,1             ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28     18.0                          CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0000(2)
                                        ;&M8958, M8959, M8955'S

1448  4305  AF                4.0  T1RPT:  XRA     A               ;SET FOR SUBGROUP 00 TEST
1449  4306  32  D6  48         13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
1450  4309  CD  52  47         18.0          CALL    FINDIT          ;GET THE PORT NUMBER
1451  430C                                  SETLPC  T1LOOP,T1CONT
 (1)  430C  21  19  43         10.0                  LXI     H,T1LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1)  430F  22  E4  48         16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  4312  21  2A  43         10.0                  LXI     H,T1CONT                ;GET THE CONTINUE ADDRESS
 (1)  4315  22  E7  48         16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1)  4318  AF                 4.0                  XRA     A
1452
1453  4319  CD  2F  48         18.0  T1LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
1454  431C  CA  24  43         10.0          JZ      T1DE0           ;JUMP IF NO ERROR DETECTED
1455  431F                                  ERRB    XDLOOP,T1DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  431F  CD  12  28         18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0001                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4322  01                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4323  04                                      .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  4324  CD  15  28         18.0          T1DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4327  DA  89  48         10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1456                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
1457
1458  432A  CD  A4  48         18.0  T1CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
1459  432D  C2  19  43         10.0          JNZ     T1LOOP          ;LOOP TILL 5 BYTES CHECKED
1460  4330                                  ENDTST  T1RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4330                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4330  CD  06  28         18.0                  CALL    REQST
```

```
(2)  4333  00                                          .BYTE                    ;DATA PATTERN NUMBER
(2)  4334  00  00                                      .WORD                    ;SYSTEM '*' COUNT
(2)  4336  00  00                                      .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2)  4338  00                                          .BYTE                    ;DATA COMPARE FLAG IF =1
(2)  4339  07                                          .BYTE   7                ;REQUEST CODE
(1)  433A  3A  9A  4F      13.0        LDA     ITERA           ;GET ITERATION COUNT
(1)  433D  3D                  4.0     DCR     A               ;DOWNCOUNT
(1)  433E  32  9A  4F      13.0        STA     ITERA           ;SAVE COUNT
(1)  4341  F2  05  43      10.0        JP      T1RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - XMC - GCR 4X5 TRANSLATION SUBGROUP 01 TEST
4344                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
4344                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0001(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 11011(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
4344                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1507                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1508                                    ;*      :       :       ENDDO
1509                                    ;*      :       :       IF TIMEOUT COUNT = 0
1510                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1511                                    ;*      :       :       :       ELSE - CONTINUE
1512                                    ;*      :       :       ENDIF
1513                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1514                                    ;*      :       ENDDO
1515                                    ;*      :       BGNDO
1516                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1517                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1518                                    ;*      :       :       :       THEN - CONTINUE
1519                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1520                                    ;*      :       :       ENDIF
1521                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1522                                    ;*      :       :       BGNDO
1523                                    ;*      :       :       :       CLOCK THE SYSTEM
1524                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1525                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1526                                    ;*      :       :       ENDDO
1527                                    ;*      :       :       IF 'WCLK' IS SET
1528                                    ;*      :       :       :       THEN - CONTINUE
1529                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1530                                    ;*      :       :       ENDIF
1531                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1532                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1533                                    ;*      :       ENDDO
1534                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1535                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1536                                    ;*      ENDDO
1537                                    ;*ENDTST
1538  4344                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1539                                    ;*XMC5 MICRO TEST 02
1540                                    ;*XMC5 MICRO ERROR 02
1541                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
1542                                    ;*M8958, M8959, M8955'S
1543                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1544                                    ;*ACTUAL = NNNN
1545                                    ;*EXPECTED = NNNN
1546                                    ;*TRANSITION COUNT = LLL
1547                                    ;*
1548                                    ;*XMC5 MICRO TEST 02
1549                                    ;*XMC5 MICRO ERROR 21
1550                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
1551                                    ;*M8958, M8959, M8955'S
1552                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1553                                    ;*FATAL ERROR - TEST ABORTED
1554                                    ;*
1555                                    ;*XMC5 MICRO TEST 02
1556                                    ;*XMC5 MICRO ERROR 22
1557                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 02
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
                                        S
                                        ; ***************************************************************************************
                                        TEST2:  TESTX   2
 (1) 4344 3E 02           7.0                           MVI     A,2             ;DEFINE THE TEST NUMBER
 (1) 4346 CD 03 28       18.0                           CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0001(2)
                                        ;&M8958, M8959, M8955'S

 4349 3E 01              7.0  T2RPT:  MVI     A,1             ;SET FOR SUBGROUP 01 TEST
 434B 32 D6 48          13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
 434E CD 52 47          18.0         CALL    FINDIT          ;GET THE PORT NUMBER
 4351                                SETLPC  T2LOOP,T2CONT
 (1) 4351 21 5E 43      10.0                 LXI     H,T2LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1) 4354 22 E4 48      16.0                 SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 4357 21 6F 43      10.0                 LXI     H,T2CONT                ;GET THE CONTINUE ADDRESS
 (1) 435A 22 E7 48      16.0                 SHLD    ERCADR+1        ;SAVE IT
 (1) 435D AF             4.0                 XRA     A

 435E CD 2F 48          18.0  T2LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
 4361 CA 69 43          10.0         JZ      T2DE0           ;JUMP IF NO ERROR DETECTED
 4364                                ERRB    XDLOOP,T2DE0,4
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 4364 CD 12 28       18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0002                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4367 02                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4368 04                                  .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 4369 CD 15 28       18.0         T2DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 436C DA 89 48       10.0                 JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

 436F CD A4 48          18.0  T2CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
 4372 C2 5E 43          10.0         JNZ     T2LOOP          ;LOOP TILL 5 BYTES CHECKED
 4375                                ENDTST  T2RPT
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 4375                            REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4375 CD 06 28       18.0                 CALL    REQST
 (2) 4378 00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2) 4379 00 00                               .WORD                   ;SYSTEM "" COUNT
 (2) 437B 00 00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 437D 00                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 437E 07                                  .BYTE   7               ;REQUEST CODE
 (1) 437F 3A 9A 4F       13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1) 4382 3D              4.0         DCR     A               ;DOWNCOUNT
 (1) 4383 32 9A 4F       13.0         STA     ITERA           ;SAVE COUNT
 (1) 4386 F2 49 43       10.0         JP      T2RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 3 - XMC - GCR 4X5 TRANSLATION SUBGROUP 02 TEST
4389                            ST
                                ;*********************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
4389                            SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0010(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10010(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
4389                            SP
                                ;*********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1631                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1632                                    ;*      :       :       ENDDO
1633                                    ;*      :       :       IF TIMEOUT COUNT = 0
1634                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1635                                    ;*      :       :       :       ELSE - CONTINUE
1636                                    ;*      :       :       ENDIF
1637                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1638                                    ;*      :       ENDDO
1639                                    ;*      :       BGNDO
1640                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1641                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1642                                    ;*      :       :       :       THEN - CONTINUE
1643                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1644                                    ;*      :       :       ENDIF
1645                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1646                                    ;*      :       :       BGNDO
1647                                    ;*      :       :       :       CLOCK THE SYSTEM
1648                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1649                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1650                                    ;*      :       :       ENDDO
1651                                    ;*      :       :       IF 'WCLK' IS SET
1652                                    ;*      :       :       :       THEN - CONTINUE
1653                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1654                                    ;*      :       :       ENDIF
1655                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1656                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1657                                    ;*      :       ENDDO
1658                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1659                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1660                                    ;*      ENDDO
1661                                    ;*ENDTST
1662  4389                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1663                                    ;*XMC5 MICRO TEST 03
1664                                    ;*XMC5 MICRO ERROR 03
1665                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
1666                                    ;*M8958, M8959, M8955'S
1667                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1668                                    ;*ACTUAL = NNNN
1669                                    ;*EXPECTED = NNNN
1670                                    ;*TRANSITION COUNT = LLL
1671                                    ;*
1672                                    ;*XMC5 MICRO TEST 03
1673                                    ;*XMC5 MICRO ERROR 21
1674                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
1675                                    ;*M8958, M8959, M8955'S
1676                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1677                                    ;*FATAL ERROR - TEST ABORTED
1678                                    ;*
1679                                    ;*XMC5 MICRO TEST 03
1680                                    ;*XMC5 MICRO ERROR 22
1681                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 03
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1691 4389                               S
 (1)                                    ; ***************************************************************************
1692 4389                               TEST3:  TESTX   3
 (1) 4389 3E 03              7.0                         MVI     A,3             ;DEFINE THE TEST NUMBER
 (1) 438B CD 03 28          18.0                         CALL    TSET            ;SETUP THE TEST
1693                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0010(2)
1694                                    ;&M8958, M8959, M8955'S
1695
1696 438E 3E 02              7.0 T3RPT:  MVI     A,2             ;SET FOR SUBGROUP 02 TEST
1697 4390 32 D6 48          13.0        STA     XFRNUM          ;SAVE THE XFR NUMBER
1698 4393 CD 52 47          18.0        CALL    FINDIT          ;GET THE PORT NUMBER
1699 4396                               SETLPC  T3LOOP,T3CONT
 (1) 4396 21 A3 43          10.0                LXI     H,T3LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1) 4399 22 E4 48          16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 439C 21 B4 43          10.0                LXI     H,T3CONT                ;GET THE CONTINUE ADDRESS
 (1) 439F 22 E7 48          16.0                SHLD    ERCADR+1        ;SAVE IT
 (1) 43A2 AF                 4.0                XRA     A
1700
1701 43A3 CD 2F 48          18.0 T3LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
1702 43A6 CA AE 43          10.0        JZ      T3DE0           ;JUMP IF NO ERROR DETECTED
1703 43A9                               ERRB    XDLOOP,T3DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1) 43A9 CD 12 28          18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0003                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 43AC 03                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 43AD 04                                 .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 43AE CD 15 28          18.0 T3DE0::     CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 43B1 DA 89 48          10.0             JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1704                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
1705
1706 43B4 CD A4 48          18.0 T3CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
1707 43B7 C2 A3 43          10.0        JNZ     T3LOOP          ;LOOP TILL 5 BYTES CHECKED
1708 43BA                               ENDTST  T3RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 43BA                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 43BA CD 06 28          18.0                        CALL    REQST
 (2) 43BD 00                                 .BYTE                   ;DATA PATTERN NUMBER
 (2) 43BE 00 00                              .WORD                   ;SYSTEM '*' COUNT
 (2) 43C0 00 00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 43C2 00                                 .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 43C3 07                                 .BYTE   7               ;REQUEST CODE
 (1) 43C4 3A 9A 4F          13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1) 43C7 3D                 4.0        DCR     A               ;DOWNCOUNT
 (1) 43C8 32 9A 4F          13.0        STA     ITERA           ;SAVE COUNT
 (1) 43CB F2 8E 43          10.0        JP      T3RPT           ;DO TEST UNTIL TILL = 0
```

```
1710                                    .SBTTL  TEST 4 - XMC - GCR 4X5 TRANSLATION SUBGROUP 03 TEST
1711  43CE                              ST
 (1)                                    ;*****************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1712                                    ;*XMC - GCR 4 X 5 TEST
1713  43CE                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1714                                    ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
1715                                    ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
1716                                    ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
1717                                    ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0011(2) TO TEST TRANSLATION ROM
1718                                    ;*PATTERN 10011(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
1719                                    ;*
1720                                    ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
1721                                    ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
1722                                    ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
1723                                    ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
1724  43CE                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1725                                    ;*BGNTST
1726                                    ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
1727                                    ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
1728                                    ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
1729                                    ;*      :       ELSE - CONTINUE
1730                                    ;*      ENDIF
1731                                    ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
1732                                    ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
1733                                    ;*      BGNDO
1734                                    ;*      :       INIT THE INTERNAL COUNTER REGISTERS
1735                                    ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
1736                                    ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
1737                                    ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
1738                                    ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
1739                                    ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
1740                                    ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
1741                                    ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
1742                                    ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
1743                                    ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
1744                                    ;*      :       BGNDO
1745                                    ;*      :       :       CLOCK THE SYSTEM
1746                                    ;*      :       :       DECREMENT THE LOOP COUNT
1747                                    ;*      :       :       DO UNTIL LOOP COUNTER = 0
1748                                    ;*      :       ENDDO
1749                                    ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
1750                                    ;*      :       BGNDO
1751                                    ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
1752                                    ;*      :       :       BGNDO
1753                                    ;*      :       :       :       CLOCK THE SYSTEM
1754                                    ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1755                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1756                                    ;*      :       :       ENDDO
1757                                    ;*      :       :       IF TIMEOUT COUNT = 0
1758                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1759                                    ;*      :       :       :       ELSE - CONTINUE
1760                                    ;*      :       :       ENDIF
1761                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1762                                    ;*      :       ENDDO
1763                                    ;*      :       BGNDO
1764                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1765                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1766                                    ;*      :       :       :       THEN - CONTINUE
1767                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1768                                    ;*      :       :       ENDIF
1769                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1770                                    ;*      :       :       BGNDO
1771                                    ;*      :       :       :       CLOCK THE SYSTEM
1772                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1773                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1774                                    ;*      :       :       ENDDO
1775                                    ;*      :       :       IF 'WCLK' IS SET
1776                                    ;*      :       :       :       THEN - CONTINUE
1777                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1778                                    ;*      :       :       ENDIF
1779                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1780                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1781                                    ;*      :       ENDDO
1782                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1783                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1784                                    ;*      ENDDO
1785                                    ;*ENDTST
1786  43CE                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1787                                    ;*XMC5 MICRO TEST 04
1788                                    ;*XMC5 MICRO ERROR 04
1789                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
1790                                    ;*M8958, M8959, M8955'S
1791                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1792                                    ;*ACTUAL = NNNN
1793                                    ;*EXPECTED = NNNN
1794                                    ;*TRANSITION COUNT = LLL
1795                                    ;*
1796                                    ;*XMC5 MICRO TEST 04
1797                                    ;*XMC5 MICRO ERROR 21
1798                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
1799                                    ;*M8958, M8959, M8955'S
1800                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1801                                    ;*FATAL ERROR - TEST ABORTED
1802                                    ;*
1803                                    ;*XMC5 MICRO TEST 04
1804                                    ;*XMC5 MICRO ERROR 22
1805                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 04
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1815  43CE                              S
 (1)                                    ; ************************************************************************************
1816  43CE                              TEST4:  TESTX   4
 (1)  43CE  3E  04            7.0                    MVI     A,4             ;DEFINE THE TEST NUMBER
 (1)  43D0  CD  03  28       18.0                    CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0011(2)
                                        ;&M8958, M8959, M8955'S
1820  43D3  3E  03            7.0  T4RPT:  MVI     A,3             ;SET FOR SUBGROUP 03 TEST
1821  43D5  32  D6  48         13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
1822  43D8  CD  52  47         18.0         CALL    FINDIT          ;GET THE PORT NUMBER
1823  43DB                               SETLPC  T4LOOP,T4CONT
 (1)  43DB  21  E8  43         10.0                 LXI     H,T4LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1)  43DE  22  E4  48         16.0                 SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  43E1  21  F9  43         10.0                 LXI     H,T4CONT                ;GET THE CONTINUE ADDRESS
 (1)  43E4  22  E7  48         16.0                 SHLD    ERCADR+1        ;SAVE IT
 (1)  43E7  AF                  4.0                 XRA     A
1824
1825  43E8  CD  2F  48         18.0  T4LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
1826  43EB  CA  F3  43         10.0         JZ      T4DEO           ;JUMP IF NO ERROR DETECTED
1827  43EE                               ERRB    XDLOOP,T4DEO,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43EE  CD  12  28         18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43F1  04                        .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  43F2  04                        .BYTE   4                       ;PRINT ROUTINE NUMBER
 (1)  43F3  CD  15  28         18.0  T4DEO:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43F6  DA  89  48         10.0         JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1828                                ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
1829
1830  43F9  CD  A4  48         18.0  T4CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
1831  43FC  C2  E8  43         10.0         JNZ     T4LOOP          ;LOOP TILL 5 BYTES CHECKED
1832  43FF                               ENDTST  T4RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43FF                              REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43FF  CD  06  28         18.0                 CALL    REQST
 (2)  4402  00                        .BYTE                           ;DATA PATTERN NUMBER
 (2)  4403  00  00                    .WORD                           ;SYSTEM '*' COUNT
 (2)  4405  00  00                    .WORD                           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4407  00                        .BYTE                           ;DATA COMPARE FLAG IF =1
 (2)  4408  07                        .BYTE   7                       ;REQUEST CODE
 (1)  4409  3A  9A  4F         13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  440C  3D                  4.0         DCR     A               ;DOWNCOUNT
 (1)  440D  32  9A  4F         13.0         STA     ITERA           ;SAVE COUNT
 (1)  4410  F2  D3  43         10.0         JP      T4RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 5 - XMC - GCR 4X5 TRANSLATION SUBGROUP 04 TEST
1835  4413                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
1837  4413                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0100(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 11101(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
1848  4413                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1879                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1880                                    ;*      :       :       ENDDO
1881                                    ;*      :       :       IF TIMEOUT COUNT = 0
1882                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1883                                    ;*      :       :       :       ELSE - CONTINUE
1884                                    ;*      :       :       ENDIF
1885                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1886                                    ;*      :       ENDDO
1887                                    ;*      :       BGNDO
1888                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1889                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1890                                    ;*      :       :       :       THEN - CONTINUE
1891                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1892                                    ;*      :       :       ENDIF
1893                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1894                                    ;*      :       :       BGNDO
1895                                    ;*      :       :       :       CLOCK THE SYSTEM
1896                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1897                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1898                                    ;*      :       :       ENDDO
1899                                    ;*      :       :       IF 'WCLK' IS SET
1900                                    ;*      :       :       :       THEN - CONTINUE
1901                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1902                                    ;*      :       :       ENDIF
1903                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1904                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1905                                    ;*      :       ENDDO
1906                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1907                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1908                                    ;*      ENDDO
1909                                    ;*ENDTST
1910  4413                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1911                                    ;*XMC5 MICRO TEST 05
1912                                    ;*XMC5 MICRO ERROR 05
1913                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
1914                                    ;*M8958, M8959, M8955'S
1915                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1916                                    ;*ACTUAL = NNNN
1917                                    ;*EXPECTED = NNNN
1918                                    ;*TRANSITION COUNT = LLL
1919                                    ;*
1920                                    ;*XMC5 MICRO TEST 05
1921                                    ;*XMC5 MICRO ERROR 21
1922                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
1923                                    ;*M8958, M8959, M8955'S
1924                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1925                                    ;*FATAL ERROR - TEST ABORTED
1926                                    ;*
1927                                    ;*XMC5 MICRO TEST 05
1928                                    ;*XMC5 MICRO ERROR 22
1929                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 05
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1939  4413                              S
 (1)                                    ; ************************************************************************************
1940  4413                              TEST5:  TESTX   5
 (1)  4413  3E  05            7.0                MVI     A,5             ;DEFINE THE TEST NUMBER
 (1)  4415  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1941                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0100(2)
1942                                    ;&M8958, M8959, M8955'S
1943
1944  4418  3E  04            7.0  T5RPT:  MVI     A,4             ;SET FOR SUBGROUP 04 TEST
1945  441A  32  D6  48         13.0        STA     XFRNUM          ;SAVE THE XFR NUMBER
1946  441D  CD  52  47         18.0        CALL    FINDIT          ;GET THE PORT NUMBER
1947  4420                              SETLPC  T5LOOP,T5CONT
 (1)  4420  21  2D  44         10.0                LXI     H,T5LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4423  22  E4  48         16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  4426  21  3E  44         10.0                LXI     H,T5CONT                ;GET THE CONTINUE ADDRESS
 (1)  4429  22  F7  48         16.0                SHLD    ERCADR+1        ;SAVE IT
 (1)  442C  AF                  4.0                XRA     A
1948
1949  442D  CD  2F  48         18.0  T5LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
1950  4430  CA  38  44         10.0        JZ      T5DE0           ;JUMP IF NO ERROR DETECTED
1951  4433                              ERRB    XDLOOP,T5DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4433  CD  12  28         18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4436  05                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4437  04                               .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  4438  CD  15  28         18.0  T5DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4438  DA  89  48         10.0                JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1952                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
1953
1954  443E  CD  A4  48         18.0  T5CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
1955  4441  C2  2D  44         10.0        JNZ     T5LOOP          ;LOOP TILL 5 BYTES CHECKED
1956  4444                              ENDTST  T5RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4444                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4444  CD  06  28         18.0                CALL    REQST
 (2)  4447  00                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  4448  00  00                           .WORD                   ;SYSTEM '*' COUNT
 (2)  444A  00  00                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  444C  00                               .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  444D  07                               .BYTE   7               ;REQUEST CODE
 (1)  444E  3A  9A  4F         13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  4451  3D                  4.0        DCR     A               ;DOWNCOUNT
 (1)  4452  32  9A  4F         13.0        STA     ITERA           ;SAVE COUNT
 (1)  4455  F2  18  44         10.0        JP      T5RPT           ;DO TEST UNTIL TILL = 0
```

```
1958                                .SBTTL  TEST 6 - XMC - GCR 4X5 TRANSLATION SUBGROUP 05 TEST
1959  4458                          ST
 (1)                                ;*********************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1960                                ;*XMC - GCR 4 X 5 TEST
1961  4458                          SD
 (1)                                ;*********************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1962                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
1963                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
1964                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
1965                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0101(2) TO TEST TRANSLATION ROM
1966                                ;*PATTERN 10101(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
1967                                ;*
1968                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
1969                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
1970                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
1971                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
1972  4458                          SP
 (1)                                ;*********************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1973                                ;*BGNTST
1974                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
1975                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
1976                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
1977                                ;*      :       ELSE - CONTINUE
1978                                ;*      ENDIF
1979                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
1980                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
1981                                ;*      BGNDO
1982                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
1983                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
1984                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
1985                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
1986                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
1987                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
1988                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
1989                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
1990                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
1991                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
1992                                ;*      :       BGNDO
1993                                ;*      :       :       CLOCK THE SYSTEM
1994                                ;*      :       :       DECREMENT THE LOOP COUNT
1995                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
1996                                ;*      :       ENDDO
1997                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
1998                                ;*      :       BGNDO
1999                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
2000                                ;*      :       :       BGNDO
2001                                ;*      :       :       :       CLOCK THE SYSTEM
2002                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2003                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2004                                    ;*      :       :       ENDDO
2005                                    ;*      :       :       IF TIMEOUT COUNT = 0
2006                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2007                                    ;*      :       :       :       ELSE - CONTINUE
2008                                    ;*      :       :       ENDIF
2009                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2010                                    ;*      :       ENDDO
2011                                    ;*      :       BGNDO
2012                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2013                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2014                                    ;*      :       :       :       THEN - CONTINUE
2015                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2016                                    ;*      :       :       ENDIF
2017                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2018                                    ;*      :       :       BGNDO
2019                                    ;*      :       :       :       CLOCK THE SYSTEM
2020                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2021                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2022                                    ;*      :       :       ENDDO
2023                                    ;*      :       :       IF 'WCLK' IS SET
2024                                    ;*      :       :       :       THEN - CONTINUE
2025                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2026                                    ;*      :       :       ENDIF
2027                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2028                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2029                                    ;*      :       ENDDO
2030                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2031                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2032                                    ;*      ENDDO
2033                                    ;*ENDTST
2034  4458                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2035                                    ;*XMC5 MICRO TEST 06
2036                                    ;*XMC5 MICRO ERROR 06
2037                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
2038                                    ;*M8958, M8959, M8955'S
2039                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2040                                    ;*ACTUAL = NNNN
2041                                    ;*EXPECTED = NNNN
2042                                    ;*TRANSITION COUNT = LLL
2043                                    ;*
2044                                    ;*XMC5 MICRO TEST 06
2045                                    ;*XMC5 MICRO ERROR 21
2046                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
2047                                    ;*M8958, M8959, M8955'S
2048                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2049                                    ;*FATAL ERROR - TEST ABORTED
2050                                    ;*
2051                                    ;*XMC5 MICRO TEST 06
2052                                    ;*XMC5 MICRO ERROR 22
2053                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 06
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
4458                                    S
 (1)                                    ; ***************************************************************************
4458                                    TEST6:  TESTX   6
 (1) 4458  3E  06            7.0                     MVI     A,6             ;DEFINE THE TEST NUMBER
 (1) 445A  CD  03  28       18.0                     CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0101(2)
                                        ;&M8958, M8959, M8955'S

445D  3E  05                7.0 T6RPT:  MVI     A,5             ;SET FOR SUBGROUP 05 TEST
445F  32  D6  48           13.0        STA     XFRNUM          ;SAVE THE XFR NUMBER
4462  CD  52  47           18.0        CALL    FINDIT          ;GET THE PORT NUMBER
4465                                    SETLPC  T6LOOP,T6CONT
 (1) 4465  21  72  44       10.0                LXI     H,T6LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1) 4468  22  E4  48       16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 446B  21  83  44       10.0                LXI     H,T6CONT                ;GET THE CONTINUE ADDRESS
 (1) 446E  22  E7  48       16.0                SHLD    ERCADR+1        ;SAVE IT
 (1) 4471  AF                4.0                XRA     A

4472  CD  2F  48           18.0 T6LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4475  CA  7D  44           10.0        JZ      T6DE0           ;JUMP IF NO ERROR DETECTED
4478                                    ERRB    XDLOOP,T6DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1) 4478  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0006                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 447B  06                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 447C  04                               .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 447D  CD  15  28       18.0        T6DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4480  DA  89  48       10.0                JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

4483  CD  A4  48           18.0 T6CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4486  C2  72  44           10.0        JNZ     T6LOOP          ;LOOP TILL 5 BYTES CHECKED
4489                                    ENDTST  T6RPT
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4489                                           CALL    REQST
 (2) 4489  CD  06  28       18.0                    .BYTE                   ;DATA PATTERN NUMBER
 (2) 448C  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2) 448D  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 448F  00  00                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 4491  00                                       .BYTE   7               ;REQUEST CODE
 (2) 4492  07
 (1) 4493  3A  9A  4F       13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1) 4496  3D                4.0        DCR     A               ;DOWNCOUNT
 (1) 4497  32  9A  4F       13.0        STA     ITERA           ;SAVE COUNT
 (1) 449A  F2  5D  44       10.0        JP      T6RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 7 - XMC - GCR 4X5 TRANSLATION SUBGROUP 06 TEST
449D                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
449D                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0110(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10110(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
449D                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2127                                    ;*       :        :        :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2128                                    ;*       :        :        :ENDDO
2129                                    ;*       :        :        :IF TIMEOUT COUNT = 0
2130                                    ;*       :        :        :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2131                                    ;*       :        :        :       ELSE - CONTINUE
2132                                    ;*       :        :        :ENDIF
2133                                    ;*       :        :        :DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2134                                    ;*       :        :    ENDDO
2135                                    ;*       :        :    BGNDO
2136                                    ;*       :        :        :GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2137                                    ;*       :        :        :IF EXPECTED DATA = ACTUAL DATA FROM XMC
2138                                    ;*       :        :        :       THEN - CONTINUE
2139                                    ;*       :        :        :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2140                                    ;*       :        :        :ENDIF
2141                                    ;*       :        :        :SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2142                                    ;*       :        :        :BGNDO
2143                                    ;*       :        :        :       CLOCK THE SYSTEM
2144                                    ;*       :        :        :       DECREMENT THE LOOP COUNTER
2145                                    ;*       :        :        :       DO UNTIL LOOP COUNTER = 0
2146                                    ;*       :        :        :ENDDO
2147                                    ;*       :        :        :IF 'WCLK' IS SET
2148                                    ;*       :        :        :       THEN - CONTINUE
2149                                    ;*       :        :        :       ELSE - REPORT IMPROPER TRANSITION RATE
2150                                    ;*       :        :        :ENDIF
2151                                    ;*       :        :        :POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2152                                    ;*       :        :        :DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2153                                    ;*       :        :    ENDDO
2154                                    ;*       :        :    POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2155                                    ;*       :        :    DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2156                                    ;*       :    ENDDO
2157                                    ;*ENDTST
2158  449D                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2159                                    ;*XMC5 MICRO TEST 07
2160                                    ;*XMC5 MICRO ERROR 07
2161                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
2162                                    ;*M8958, M8959, M8955'S
2163                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2164                                    ;*ACTUAL = NNNN
2165                                    ;*EXPECTED = NNNN
2166                                    ;*TRANSITION COUNT = LLL
2167                                    ;*
2168                                    ;*XMC5 MICRO TEST 07
2169                                    ;*XMC5 MICRO ERROR 21
2170                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
2171                                    ;*M8958, M8959, M8955'S
2172                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2173                                    ;*FATAL ERROR - TEST ABORTED
2174                                    ;*
2175                                    ;*XMC5 MICRO TEST 07
2176                                    ;*XMC5 MICRO ERROR 22
2177                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 07
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
                                        S
                                        ; ***************************************************************************
                                        TEST7:  TESTX   7
                                                        MVI     A,7             ;DEFINE THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0110(2)
                                        ;&M8958, M8959, M8955'S

                                        T7RPT:  MVI     A,6             ;SET FOR SUBGROUP 06 TEST
                                                STA     XFRNUM          ;SAVE THE XFR NUMBER
                                                CALL    FINDIT          ;GET THE PORT NUMBER
                                                SETLPC  T7LOOP,T7CONT
                                                        LXI     H,T7LOOP                ;GET THE LOOP-ON-ERR ADDRESS
                                                        SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
                                                        LXI     H,T7CONT                ;GET THE CONTINUE ADDRESS
                                                        SHLD    ERCADR+1        ;SAVE IT
                                                        XRA     A

                                        T7LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
                                                JZ      T7DE0           ;JUMP IF NO ERROR DETECTED
                                                ERRB    XDLOOP,T7DE0,4
                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE   4               ;PRINT ROUTINE NUMBER
                                        T7DE0::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

                                        T7CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
                                                JNZ     T7LOOP          ;LOOP TILL 5 BYTES CHECKED
                                                ENDTST  T7RPT
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
                                                        CALL    REQST
                                                        .BYTE                   ;DATA PATTERN NUMBER
                                                        .WORD                   ;SYSTEM '*' COUNT
                                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                        .BYTE                   ;DATA COMPARE FLAG IF =1
                                                        .BYTE   7               ;REQUEST CODE
                                                LDA     ITERA           ;GET ITERATION COUNT
                                                DCR     A               ;DOWNCOUNT
                                                STA     ITERA           ;SAVE COUNT
                                                JP      T7RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - XMC - GCR 4X5 TRANSLATION SUBGROUP 07 TEST
44E2                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
44E2                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0111(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10111(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
44E2                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2251                                    ;*       :        :        :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2252                                    ;*       :        :       ENDDO
2253                                    ;*       :        :       IF TIMEOUT COUNT = 0
2254                                    ;*       :        :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2255                                    ;*       :        :       :       ELSE - CONTINUE
2256                                    ;*       :        :       ENDIF
2257                                    ;*       :        :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2258                                    ;*       :       ENDDO
2259                                    ;*       :       BGNDO
2260                                    ;*       :        :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2261                                    ;*       :        :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2262                                    ;*       :        :       :       THEN - CONTINUE
2263                                    ;*       :        :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2264                                    ;*       :        :       ENDIF
2265                                    ;*       :        :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2266                                    ;*       :        :       BGNDO
2267                                    ;*       :        :       :       CLOCK THE SYSTEM
2268                                    ;*       :        :       :       DECREMENT THE LOOP COUNTER
2269                                    ;*       :        :       :       DO UNTIL LOOP COUNTER = 0
2270                                    ;*       :        :       ENDDO
2271                                    ;*       :        :       IF 'WCLK' IS SET
2272                                    ;*       :        :       :       THEN - CONTINUE
2273                                    ;*       :        :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2274                                    ;*       :        :       ENDIF
2275                                    ;*       :        :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2276                                    ;*       :        :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2277                                    ;*       :       ENDDO
2278                                    ;*       :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2279                                    ;*       :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2280                                    ;*      ENDDO
2281                                    ;*ENDTST
2282  44E2                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2283                                    ;*XMC5 MICRO TEST 10
2284                                    ;*XMC5 MICRO ERROR 10
2285                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
2286                                    ;*M8958, M8959, M8955'S
2287                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2288                                    ;*ACTUAL = NNNN
2289                                    ;*EXPECTED = NNNN
2290                                    ;*TRANSITION COUNT = LLL
2291                                    ;*
2292                                    ;*XMC5 MICRO TEST 10
2293                                    ;*XMC5 MICRO ERROR 21
2294                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
2295                                    ;*M8958, M8959, M8955'S
2296                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2297                                    ;*FATAL ERROR - TEST ABORTED
2298                                    ;*
2299                                    ;*XMC5 MICRO TEST 10
2300                                    ;*XMC5 MICRO ERROR 22
2301                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
```

```
                                ;*M8958, M8959, M8955'S
                                ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                ;*FATAL ERROR - TEST ABORTED!
                                ;*
                                ;*XMC5 MICRO TEST 10
                                ;*XMC5 MICRO ERROR 23
                                ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
                                ;*M8958, M8959, M8955'S
                                ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
                                ;
 44E2                           ; ***************************************************************************
  (1)                           ;
 44E2                           TEST10: TESTX   @10
 44E2  3E  08          7.0                       MVI     A,@10           ;DEFINE THE TEST NUMBER
 44E4  CD  03  28     18.0                       CALL    TSET            ;SETUP THE TEST
                                ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0111(2)
                                ;&M8958, M8959, M8955'S
 44E7  3E  07          7.0   T10RPT: MVI     A,7             ;SET FOR SUBGROUP 07 TEST
 44E9  32  D6  48     13.0           STA     XFRNUM          ;SAVE THE XFR NUMBER
 44EC  CD  52  47     18.0           CALL    FINDIT          ;GET THE PORT NUMBER
 44EF                               SETLPC  T10LOOP,T10CONT
 44EF  21  FC  44     10.0                   LXI     H,T10LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 44F2  22  E4  48     16.0                   SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 44F5  21  0D  45     10.0                   LXI     H,T10CONT               ;GET THE CONTINUE ADDRESS
 44F8  22  E7  48     16.0                   SHLD    ERCADR+1        ;SAVE IT
 44FB  AF              4.0                   XRA     A
 44FC  CD  2F  48     18.0   T10LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
 44FF  CA  07  45     10.0           JZ      T10DE0          ;JUMP IF NO ERROR DETECTED
 4502                               ERRB    XDLOOP,T10DE0,4
  (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 4502  CD  12  28     18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)     0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 4505  08                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 4506  04                                   .BYTE   4               ;PRINT ROUTINE NUMBER
 4507  CD  15  28     18.0           T10DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 450A  DA  89  48     10.0                   JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
 450D  CD  A4  48     18.0   T10CONT: CALL   NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
 4510  C2  FC  44     10.0           JNZ     T10LOOP         ;LOOP TILL 5 BYTES CHECKED
 4513                               ENDTST  T10RPT
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 4513                                       CALL    REQST
 4513  CD  06  28     18.0                   .BYTE                   ;DATA PATTERN NUMBER
 4516  00                                   .WORD                   ;SYSTEM '=' COUNT
 4517  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 4519  00  00                               .BYTE                   ;DATA COMPARE FLAG IF =1
 451B  00                                   .BYTE   7               ;REQUEST CODE
 451C  07
 451D  3A  9A  4F     13.0           LDA     ITERA           ;GET ITERATION COUNT
 4520  3D              4.0           DCR     A               ;DOWNCOUNT
 4521  32  9A  4F     13.0           STA     ITERA           ;SAVE COUNT
 4524  F2  E7  44     10.0           JP      T10RPT          ;DO TEST UNTIL TILL = 0
```

```
2330                                    .SBTTL  TEST 11 - XMC - GCR 4X5 TRANSLATION SUBGROUP 10 TEST
2331  4527                              ST
 (1)                                    ;*********************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
2332                                    ;*XMC - GCR 4 X 5 TEST
2333  4527                              SD
 (1)                                    ;*********************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2334                                    ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
2335                                    ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
2336                                    ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
2337                                    ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1000(2) TO TEST TRANSLATION ROM
2338                                    ;*PATTERN 11010(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
2339                                    ;*
2340                                    ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
2341                                    ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
2342                                    ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
2343                                    ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
2344  4527                              SP
 (1)                                    ;*********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2345                                    ;*BGNTST
2346                                    ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
2347                                    ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
2348                                    ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
2349                                    ;*      :       ELSE - CONTINUE
2350                                    ;*      ENDIF
2351                                    ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
2352                                    ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
2353                                    ;*      BGNDO
2354                                    ;*      :       INIT THE INTERNAL COUNTER REGISTERS
2355                                    ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
2356                                    ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
2357                                    ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
2358                                    ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
2359                                    ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
2360                                    ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
2361                                    ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
2362                                    ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
2363                                    ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
2364                                    ;*      :       BGNDO
2365                                    ;*      :       :       CLOCK THE SYSTEM
2366                                    ;*      :       :       DECREMENT THE LOOP COUNT
2367                                    ;*      :       :       DO UNTIL LOOP COUNTER = 0
2368                                    ;*      :       ENDDO
2369                                    ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
2370                                    ;*      :       BGNDO
2371                                    ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
2372                                    ;*      :       :       BGNDO
2373                                    ;*      :       :       :       CLOCK THE SYSTEM
2374                                    ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2375                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2376                    ;*      :       :       ENDDO
2377                    ;*      :       :       IF TIMEOUT COUNT = 0
2378                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2379                    ;*      :       :       :       ELSE - CONTINUE
2380                    ;*      :       :       ENDIF
2381                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2382                    ;*      :       ENDDO
2383                    ;*      :       BGNDO
2384                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2385                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2386                    ;*      :       :       :       THEN - CONTINUE
2387                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2388                    ;*      :       :       ENDIF
2389                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2390                    ;*      :       :       BGNDO
2391                    ;*      :       :       :       CLOCK THE SYSTEM
2392                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2393                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2394                    ;*      :       :       ENDDO
2395                    ;*      :       :       IF 'WCLK' IS SET
2396                    ;*      :       :       :       THEN - CONTINUE
2397                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2398                    ;*      :       :       ENDIF
2399                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2400                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2401                    ;*      :       ENDDO
2402                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2403                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2404                    ;*      ENDDO
2405                    ;*ENDTST
2406  4527              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2407                    ;*XMC5 MICRO TEST 11
2408                    ;*XMC5 MICRO ERROR 11
2409                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
2410                    ;*M8958, M8959, M8955'S
2411                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2412                    ;*ACTUAL = NNNN
2413                    ;*EXPECTED = NNNN
2414                    ;*TRANSITION COUNT = LLL
2415                    ;*
2416                    ;*XMC5 MICRO TEST 11
2417                    ;*XMC5 MICRO ERROR 21
2418                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
2419                    ;*M8958, M8959, M8955'S
2420                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2421                    ;*FATAL ERROR - TEST ABORTED
2422                    ;*
2423                    ;*XMC5 MICRO TEST 11
2424                    ;*XMC5 MICRO ERROR 22
2425                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 11
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
2435  4527                              S
 (1)                                    ; ***************************************************************************
2436  4527                              TEST11: TESTX   @11
 (1)  4527  3E  09          7.0                     MVI     A,@11           ;DEFINE THE TEST NUMBER
 (1)  4529  CD  03  28     18.0                     CALL    TSET            ;SETUP THE TEST
2437                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1000(2)
2438                                    ;&M8958, M8959, M8955'S
2439
2440  452C  3E  08          7.0  T11RPT: MVI     A,@10           ;SET FOR SUBGROUP 10 TEST
2441  452E  32  D6  48     13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
2442  4531  CD  52  47     18.0          CALL    FINDIT          ;GET THE PORT NUMBER
2443  4534                              SETLPC  T11LOOP,T11CONT
 (1)  4534  21  41  45     10.0                  LXI     H,T11LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4537  22  E4  48     16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  453A  21  52  45     10.0                  LXI     H,T11CONT               ;GET THE CONTINUE ADDRESS
 (1)  453D  22  E7  48     16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1)  4540  AF              4.0                  XRA     A
2444
2445  4541  CD  2F  48     18.0  T11LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
2446  4544  CA  4C  45     10.0          JZ      T11DE0          ;JUMP IF NO ERROR DETECTED
2447  4547                              ERRB    XDLOOP T11DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4547  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                        MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454A  09                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454B  04                          .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  454C  CD  15  28     18.0  T11DE0::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  454F  DA  89  48     10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2448                                    ;>SUBGROUP 4X5 DATA INCORREC` FROM XMC
2449
2450  4552  CD  A4  48     18.0  T11CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
2451  4555  C2  41  45     10.0          JNZ     T11LOOP         ;LOOP TILL 5 BYTES CHECKED
2452  4558                              ENDTST  T11RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4558                                      CALL    REQST
 (2)  4558  CD  06  28     18.0                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  455B  00                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  455C  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455E  00  00                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4560  00                                  .BYTE   7               ;REQUEST CODE
 (1)  4562  3A  9A  4F     13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4565  3D              4.0          DCR     A               ;DOWNCOUNT
 (1)  4566  32  9A  4F     13.0          STA     ITERA           ;SAVE COUNT
 (1)  4569  F2  2C  45     10.0          JP      T11RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - XMC - GCR 4X5 TRANSLATION SUBGROUP 11 TEST
456C                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
456C                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1001(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01001(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
456C                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2499                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2500                                    ;*      :       :       ENDDO
2501                                    ;*      :       :       IF TIMEOUT COUNT = 0
2502                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2503                                    ;*      :       :       :       ELSE - CONTINUE
2504                                    ;*      :       :       ENDIF
2505                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2506                                    ;*      :       ENDDO
2507                                    ;*      :       BGNDO
2508                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2509                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2510                                    ;*      :       :       :       THEN - CONTINUE
2511                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2512                                    ;*      :       :       ENDIF
2513                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2514                                    ;*      :       :       BGNDO
2515                                    ;*      :       :       :       CLOCK THE SYSTEM
2516                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2517                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2518                                    ;*      :       :       ENDDO
2519                                    ;*      :       :       IF 'WCLK' IS SET
2520                                    ;*      :       :       :       THEN - CONTINUE
2521                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2522                                    ;*      :       :       ENDIF
2523                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2524                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2525                                    ;*      :       ENDDO
2526                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2527                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2528                                    ;*      ENDDO
2529                                    ;*ENDTST
2530  456C                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2531                                    ;*XMC5 MICRO TEST 12
2532                                    ;*XMC5 MICRO ERROR 12
2533                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
2534                                    ;*M8958, M8959, M8955'S
2535                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2536                                    ;*ACTUAL = NNNN
2537                                    ;*EXPECTED = NNNN
2538                                    ;*TRANSITION COUNT = LLL
2539                                    ;*
2540                                    ;*XMC5 MICRO TEST 12
2541                                    ;*XMC5 MICRO ERROR 21
2542                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
2543                                    ;*M8958, M8959, M8955'S
2544                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2545                                    ;*FATAL ERROR - TEST ABORTED
2546                                    ;*
2547                                    ;*XMC5 MICRO TEST 12
2548                                    ;*XMC5 MICRO ERROR 22
2549                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 12
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
2559  456C                              S
 (1)                                    ; ***************************************************************************************
2560  456C                             TEST12: TESTX   @12
 (1)  456C  3E  0A              7.0                  MVI     A,@12           ;DEFINE THE TEST NUMBER
 (1)  456E  CD  03  28         18.0                  CALL    TSET            ;SETUP THE TEST
2561                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1001(2)
2562                                    ;&M8958, M8959, M8955'S
2563
2564  4571  3E  09                7.0  T12RPT: MVI     A,@11           ;SET FOR SUBGROUP 11 TEST
2565  4573  32  D6  48            13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
2566  4576  CD  52  47            18.0          CALL    FINDIT          ;GET THE PORT NUMBER
2567  4579                                      SETLPC  T12LOOP,T12CONT
 (1)  4579  21  86  45            10.0                  LXI     H,T12LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  457C  22  E4  48            16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  457F  21  97  45            10.0                  LXI     H,T12CONT               ;GET THE CONTINUE ADDRESS
 (1)  4582  22  E7  48            16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1)  4585  AF                     4.0                  XRA     A
2568
2569  4586  CD  2F  48            18.0  T12LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
2570  4589  CA  91  45            10.0          JZ      T12DE0          ;JUMP IF NO ERROR DETECTED
2571  458C                                      ERRB    XDLOOP,T12DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  458C  CD  12  28            18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000A                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  458F  0A                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4590  04                                          .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  4591  CD  15  28            18.0          T12DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4594  DA  89  48            10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2572                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
2573
2574  4597  CD  A4  48            18.0  T12CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
2575  459A  C2  86  45            10.0          JNZ     T12LOOP         ;LOOP TILL 5 BYTES CHECKED
2576  459D                                      ENDTST  T12RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  459D                                              CALL    REQST
 (2)  459D  CD  06  28            18.0                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  45A0  00                                          .WORD                   ;SYSTEM '*' COUNT
 (2)  45A1  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45A3  00  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  45A5  00                                          .BYTE   7               ;REQUEST CODE
 (2)  45A6  07
 (1)  45A7  3A  9A  4F            13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  45AA  3D                     4.0          DCR     A               ;DOWNCOUNT
 (1)  45AB  32  9A  4F            13.0          STA     ITERA           ;SAVE COUNT
 (1)  45AE  F2  71  45            10.0          JP      T12RPT          ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 13 - XMC - GCR 4X5 TRANSLATION SUBGROUP 12 TEST
 45B1                                   ST
                                        ;*********************************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*XMC - GCR 4 X 5 TEST
 45B1                                   SD
                                        ;*********************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                        ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                        ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                        ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1010(2) TO TEST TRANSLATION ROM
                                        ;*PATTERN 01010(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                        ;*
                                        ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                        ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                        ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                        ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
 45B1                                   SP
                                        ;*********************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                        ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                        ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                        ;*      :       ELSE - CONTINUE
                                        ;*      ENDIF
                                        ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                        ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                        ;*      BGNDO
                                        ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                        ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                        ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                        ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                        ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                        ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                        ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                        ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                        ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                        ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                        ;*      :       BGNDO
                                        ;*      :       :       CLOCK THE SYSTEM
                                        ;*      :       :       DECREMENT THE LOOP COUNT
                                        ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                        ;*      :       ENDDO
                                        ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                        ;*      :       BGNDO
                                        ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                        ;*      :       :       BGNDO
                                        ;*      :       :       :       CLOCK THE SYSTEM
                                        ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
                                ;*       :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
                                ;*       :       :       ENDDO
                                ;*       :       :       IF TIMEOUT COUNT = 0
                                ;*       :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
                                ;*       :       :       :       ELSE - CONTINUE
                                ;*       :       :       ENDIF
                                ;*       :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
                                ;*       :       ENDDO
                                ;*       :       BGNDO
                                ;*       :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
                                ;*       :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
                                ;*       :       :       :       THEN - CONTINUE
                                ;*       :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
                                ;*       :       :       ENDIF
                                ;*       :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
                                ;*       :       :       BGNDO
                                ;*       :       :       :       CLOCK THE SYSTEM
                                ;*       :       :       :       DECREMENT THE LOOP COUNTER
                                ;*       :       :       :       DO UNTIL LOOP COUNTER = 0
                                ;*       :       :       ENDDO
                                ;*       :       :       IF 'WCLK' IS SET
                                ;*       :       :       :       THEN - CONTINUE
                                ;*       :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
                                ;*       :       :       ENDIF
                                ;*       :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
                                ;*       :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
                                ;*       :       ENDDO
                                ;*       :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
                                ;*       :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
                                ;*       ENDDO
                                ;*ENDTST
45B1                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*XMC5 MICRO TEST 13
                                ;*XMC5 MICRO ERROR 13
                                ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
                                ;*M8958, M8959, M8955'S
                                ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*TRANSITION COUNT = LLL
                                ;*
                                ;*XMC5 MICRO TEST 13
                                ;*XMC5 MICRO ERROR 21
                                ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
                                ;*M8958, M8959, M8955'S
                                ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*XMC5 MICRO TEST 13
                                ;*XMC5 MICRO ERROR 22
                                ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
```

```
2674                                    ;*M8958, M8959, M8955'S
2675                                    ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
2676                                    ;*FATAL ERROR - TEST ABORTED!
2677                                    ;*
2678                                    ;*XMC5 MICRO TEST 13
2679                                    ;*XMC5 MICRO ERROR 23
2680                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
2681                                    ;*M8958, M8959, M8955'S
2682                                    ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
2683  45B1                              S
 (1)                                    ; *****************************************************************************
2684  45B1                              TEST13: TESTX   @13
 (1)  45B1  3E  0B            7.0                 MVI     A,@13           ;DEFINE THE TEST NUMBER
 (1)  45B3  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
2685                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1010(2)
2686                                    ;&M8958, M8959, M8955'S
2687
2688  45B6  3E  0A                7.0 T13RPT: MVI     A,@12           ;SET FOR SUBGROUP 12 TEST
2689  45B8  32  D6  48           13.0        STA     XFRNUM          ;SAVE THE XFR NUMBER
2690  45BB  CD  52  47           18.0        CALL    FINDIT          ;GET THE PORT NUMBER
2691  45BE                              SETLPC  T13LOOP,T13CONT
 (1)  45BE  21  CB  45           10.0                LXI     H,T13LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  45C1  22  E4  48           16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  45C4  21  DC  45           10.0                LXI     H,T13CONT               ;GET THE CONTINUE ADDRESS
 (1)  45C7  22  E7  48           16.0                SHLD    ERCADR+1        ;SAVE IT
 (1)  45CA  AF                    4.0                XRA     A
2692
2693  45CB  CD  2F  48           18.0 T13LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
2694  45CE  CA  D6  45           10.0          JZ      T13DE0          ;JUMP IF NO ERROR DETECTED
2695  45D1                              ERRB    XDLOOP,T13DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  45D1  CD  12  28           18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000B                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D4  0B                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45D5  04                        .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  45D6  CD  15  28           18.0 T13DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45D9  DA  89  48           10.0                JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2696                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
2697
2698  45DC  CD  A4  48           18.0 T13CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
2699  45DF  C2  CB  45           10.0          JNZ     T13LOOP         ;LOOP TILL 5 BYTES CHECKED
2700  45E2                              ENDTST  T13RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45E2                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45E2  CD  06  28           18.0                CALL    REQST
 (2)  45E5  00                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  45E6  00  00                    .WORD                   ;SYSTEM '"' COUNT
 (2)  45E8  00  00                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45EA  00                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  45EB  07                        .BYTE   7               ;REQUEST CODE
 (1)  45EC  3A  9A  4F           13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  45EF  3D                    4.0        DCR     A               ;DOWNCOUNT
 (1)  45F0  32  9A  4F           13.0        STA     ITERA           ;SAVE COUNT
 (1)  45F3  F2  B6  45           10.0        JP      T13RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - XMC - GCR 4X5 TRANSLATION SUBGROUP 13 TEST
45F6                            ST
                                ;*********************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
45F6                            SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1011(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01011(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
45F6                            SP
                                ;*********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2747                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2748                                    ;*      :       :       ENDDO
2749                                    ;*      :       :       IF TIMEOUT COUNT = 0
2750                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2751                                    ;*      :       :       :       ELSE - CONTINUE
2752                                    ;*      :       :       ENDIF
2753                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2754                                    ;*      :       ENDDO
2755                                    ;*      :       BGNDO
2756                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2757                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2758                                    ;*      :       :       :       THEN - CONTINUE
2759                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2760                                    ;*      :       :       ENDIF
2761                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2762                                    ;*      :       :       BGNDO
2763                                    ;*      :       :       :       CLOCK THE SYSTEM
2764                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2765                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2766                                    ;*      :       :       ENDDO
2767                                    ;*      :       :       IF 'WCLK' IS SET
2768                                    ;*      :       :       :       THEN - CONTINUE
2769                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2770                                    ;*      :       :       ENDIF
2771                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2772                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2773                                    ;*      :       ENDDO
2774                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2775                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2776                                    ;*      ENDDO
2777                                    ;*ENDTST
2778  45F6                              SE
  (1)                                   ;*****************************************************************************
  (1)                                   ;*ERRORS
  (1)                                   ;*------
2779                                    ;*XMC5 MICRO TEST 14
2780                                    ;*XMC5 MICRO ERROR 14
2781                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
2782                                    ;*M8958, M8959, M8955'S
2783                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2784                                    ;*ACTUAL = NNNN
2785                                    ;*EXPECTED = NNNN
2786                                    ;*TRANSITION COUNT = LLL
2787                                    ;*
2788                                    ;*XMC5 MICRO TEST 14
2789                                    ;*XMC5 MICRO ERROR 21
2790                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
2791                                    ;*M8958, M8959, M8955'S
2792                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2793                                    ;*FATAL ERROR - TEST ABORTED
2794                                    ;*
2795                                    ;*XMC5 MICRO TEST 14
2796                                    ;*XMC5 MICRO ERROR 22
2797                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 14
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
                                        S
                                        ; ***************************************************************************
                                        TEST14: TESTX   @14
 (1) 45F6 3E 0C            7.0                 MVI     A,@14           ;DEFINE THE TEST NUMBER
 (1) 45F8 CD 03 28        18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1011(2)
                                        ;&M8958, M8959, M8955'S

 45FB 3E 0B                7.0  T14RPT: MVI     A,@13           ;SET FOR SUBGROUP 13 TEST
 45FD 32 D6 48            13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
 4600 CD 52 47            18.0         CALL    FINDIT          ;GET THE PORT NUMBER
 4603                                  SETLPC  T14LOOP,T14CONT
 (1) 4603 21 10 46        10.0                 LXI     H,T14LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1) 4606 22 E4 48        16.0                 SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 4609 21 21 46        10.0                 LXI     H,T14CONT               ;GET THE CONTINUE ADDRESS
 (1) 460C 22 E7 48        16.0                 SHLD    ERCADR+1        ;SAVE IT
 (1) 460F AF               4.0                 XRA     A

 4610 CD 2F 48            18.0  T14LOOP: CALL   XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
 4613 CA 1B 46            10.0         JZ      T14DE0          ;JUMP IF NO ERROR DETECTED
 4616                                  ERRB    XDLOOP,T14DE0,4
                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1) 4616 CD 12 28        18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000C                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4619 0C                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 461A 04                                  .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 461B CD 15 28        18.0  T14DE0::      CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 461E DA 89 48        10.0                 JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                       ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

 4621 CD A4 48            18.0  T14CONT: CALL   NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
 4624 C2 10 46            10.0         JNZ     T14LOOP         ;LOOP TILL 5 BYTES CHECKED
 4627                                  ENDTST  T14RPT
                                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4627                                      CALL    REQST
 (2) 4627 CD 06 28        18.0                 .BYTE                   ;DATA PATTERN NUMBER
 (2) 462A 00                                   .WORD                   ;SYSTEM '*' COUNT
 (2) 462B 00 00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 462D 00 00                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 462F 00                                   .BYTE   7               ;REQUEST CODE
 (2) 4630 07
 (1) 4631 3A 9A 4F        13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1) 4634 3D               4.0         DCR     A               ;DOWNCOUNT
 (1) 4635 32 9A 4F        13.0         STA     ITERA           ;SAVE COUNT
 (1) 4638 F2 FB 45        10.0         JP      T14RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 15 - XMC - GCR 4X5 TRANSLATION SUBGROUP 14 TEST
2827  463B                     ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
2829  463B                     SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1100(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 11110(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
2840  463B                     SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2871                                    ;*       :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2872                                    ;*       :       :       ENDDO
2873                                    ;*       :       :       IF TIMEOUT COUNT = 0
2874                                    ;*       :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2875                                    ;*       :       :       :       ELSE - CONTINUE
2876                                    ;*       :       :       ENDIF
2877                                    ;*       :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2878                                    ;*       :       ENDDO
2879                                    ;*       :       BGNDO
2880                                    ;*       :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2881                                    ;*       :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2882                                    ;*       :       :       :       THEN - CONTINUE
2883                                    ;*       :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2884                                    ;*       :       :       ENDIF
2885                                    ;*       :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2886                                    ;*       :       :       BGNDO
2887                                    ;*       :       :       :       CLOCK THE SYSTEM
2888                                    ;*       :       :       :       DECREMENT THE LOOP COUNTER
2889                                    ;*       :       :       :       DO UNTIL LOOP COUNTER = 0
2890                                    ;*       :       :       ENDDO
2891                                    ;*       :       :       IF 'WCLK' IS SET
2892                                    ;*       :       :       :       THEN - CONTINUE
2893                                    ;*       :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2894                                    ;*       :       :       ENDIF
2895                                    ;*       :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2896                                    ;*       :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2897                                    ;*       :       ENDDO
2898                                    ;*       :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2899                                    ;*       :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2900                                    ;*       ENDDO
2901                                    ;*ENDTST
2902  4638                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2903                                    ;*XMC5 MICRO TEST 15
2904                                    ;*XMC5 MICRO ERROR 15
2905                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
2906                                    ;*M8958, M8959, M8955'S
2907                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2908                                    ;*ACTUAL = NNNN
2909                                    ;*EXPECTED = NNNN
2910                                    ;*TRANSITION COUNT = LLL
2911                                    ;*
2912                                    ;*XMC5 MICRO TEST 15
2913                                    ;*XMC5 MICRO ERROR 21
2914                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
2915                                    ;*M8958, M8959, M8955'S
2916                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2917                                    ;*FATAL ERROR - TEST ABORTED
2918                                    ;*
2919                                    ;*XMC5 MICRO TEST 15
2920                                    ;*XMC5 MICRO ERROR 22
2921                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 15
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
463B                                    S
 (1)                                    ; ***************************************************************************
463B                                    TEST15: TESTX   @15
 (1) 463B  3E  0D            7.0                    MVI     A,@15           ;DEFINE THE TEST NUMBER
 (1) 463D  CD  03  28       18.0                    CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1100(2)
                                        ;&M8958, M8959, M8955'S

4640  3E  0C                 7.0  T15RPT: MVI     A,@14           ;SET FOR SUBGROUP 14 TEST
4642  32  D6  48            13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
4645  CD  52  47            18.0         CALL    FINDIT          ;GET THE PORT NUMBER
4648                                      SETLPC  T15LOOP,T15CONT
 (1) 4648  21  55  46        10.0                 LXI     H,T15LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1) 464B  22  E4  48        16.0                 SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 464E  21  66  46        10.0                 LXI     H,T15CONT               ;GET THE CONTINUE ADDRESS
 (1) 4651  22  E7  48        16.0                 SHLD    ERCADR+1        ;SAVE IT
 (1) 4654  AF                 4.0                 XRA     A

4655  CD  2F  48            18.0  T15LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4658  CA  60  46            10.0           JZ      T15DE0          ;JUMP IF NO ERROR DETECTED
465B                                       ERRB    XDLOOP,T15DE0,4
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1) 465B  CD  12  28        18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000D                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 465E  0D                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 465F  04                                     .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 4660  CD  15  28        18.0                 T15DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4663  DA  89  48        10.0                         JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

4666  CD  A4  48            18.0  T15CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4669  C2  55  46            10.0           JNZ     T15LOOP         ;LOOP TILL 5 BYTES CHECKED
466C                                       ENDTST  T15RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 466C                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 466C  CD  06  28        18.0                 CALL    REQST
 (2) 466F  00                                     .BYTE                   ;DATA PATTERN NUMBER
 (2) 4670  00  00                                 .WORD                   ;SYSTEM '*' COUNT
 (2) 4672  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 4674  00                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 4675  07                                     .BYTE   7               ;REQUEST CODE
 (1) 4676  3A  9A  4F        13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1) 4679  3D                 4.0                 DCR     A               ;DOWNCOUNT
 (1) 467A  32  9A  4F        13.0                 STA     ITERA           ;SAVE COUNT
 (1) 467D  F2  40  46        10.0                 JP      T15RPT          ;DO TEST UNTIL TILL = 0
```

```
2950
2951  4680                          .SBTTL  TEST 16 - XMC - GCR 4X5 TRANSLATION SUBGROUP 15 TEST
 (1)                                ST
 (1)                                ;*********************************************************************************
 (1)                                ;*TEST TITLE
                                    ;*---- -----
2952                                ;*XMC - GCR 4 X 5 TEST
2953  4680                          SD
 (1)                                ;*********************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
2954                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
2955                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
2956                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
2957                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1101(2) TO TEST TRANSLATION ROM
2958                                ;*PATTERN 01101(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
2959                                ;*
2960                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
2961                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
2962                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
2963                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
2964  4680                          SP
 (1)                                ;*********************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
2965                                ;*BGNTST
2966                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
2967                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
2968                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
2969                                ;*      :       ELSE - CONTINUE
2970                                ;*      ENDIF
2971                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
2972                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
2973                                ;*      BGNDO
2974                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
2975                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
2976                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
2977                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
2978                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
2979                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
2980                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
2981                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
2982                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
2983                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
2984                                ;*      :       BGNDO
2985                                ;*      :       :       CLOCK THE SYSTEM
2986                                ;*      :       :       DECREMENT THE LOOP COUNT
2987                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
2988                                ;*      :       ENDDO
2989                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
2990                                ;*      :       BGNDO
2991                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
2992                                ;*      :       :       BGNDO
2993                                ;*      :       :       :       CLOCK THE SYSTEM
2994                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
                        ;*        :        :        :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
                        ;*        :        :        :      ENDDO
                        ;*        :        :        :      IF TIMEOUT COUNT = 0
                        ;*        :        :        :      :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
                        ;*        :        :        :      :       ELSE - CONTINUE
                        ;*        :        :        :      ENDIF
                        ;*        :        :        :      DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
                        ;*        :        :      ENDDO
                        ;*        :        :      BGNDO
                        ;*        :        :        :      GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
                        ;*        :        :        :      IF EXPECTED DATA = ACTUAL DATA FROM XMC
                        ;*        :        :        :      :       THEN - CONTINUE
                        ;*        :        :        :      :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
                        ;*        :        :        :      ENDIF
                        ;*        :        :        :      SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
                        ;*        :        :        :      BGNDO
                        ;*        :        :        :      :       CLOCK THE SYSTEM
                        ;*        :        :        :      :       DECREMENT THE LOOP COUNTER
                        ;*        :        :        :      :       DO UNTIL LOOP COUNTER = 0
                        ;*        :        :        :      ENDDO
                        ;*        :        :        :      IF 'WCLK' IS SET
                        ;*        :        :        :      :       THEN - CONTINUE
                        ;*        :        :        :      :       ELSE - REPORT IMPROPER TRANSITION RATE
                        ;*        :        :        :      ENDIF
                        ;*        :        :        :      POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
                        ;*        :        :        :      DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
                        ;*        :        :      ENDDO
                        ;*        :        :      POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
                        ;*        :        :      DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
                        ;*        :      ENDDO
                        ;*ENDTST
                        SE
                        ;*****************************************************************************
                        ;*ERRORS
                        ;*------
                        ;*XMC5 MICRO TEST 16
                        ;*XMC5 MICRO ERROR 16
                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
                        ;*M8958, M8959, M8955'S
                        ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*TRANSITION COUNT = LLL
                        ;*
                        ;*XMC5 MICRO TEST 16
                        ;*XMC5 MICRO ERROR 21
                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
                        ;*M8958, M8959, M8955'S
                        ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
                        ;*FATAL ERROR - TEST ABORTED
                        ;*
                        ;*XMC5 MICRO TEST 16
                        ;*XMC5 MICRO ERROR 22
                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
```

```
3046                                    ;*M8958, M8959, M8955'S
3047                                    ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
3048                                    ;*FATAL ERROR - TEST ABORTED!
3049                                    ;*
3050                                    ;*XMC5 MICRO TEST 16
3051                                    ;*XMC5 MICRO ERROR 23
3052                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
3053                                    ;*M8958, M8959, M8955'S
3054                                    ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
3055  4680                              S
 (1)                                    ; ***************************************************************************
3056  4680                              TEST16: TESTX   @16
 (1)  4680  3E  0E           7.0                  MVI     A,@16           ;DEFINE THE TEST NUMBER
 (1)  4682  CD  03  28      18.0                  CALL    TSET            ;SETUP THE TEST
3057                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1101(2)
3058                                    ;&M8958, M8959, M8955'S
3059
3060  4685  3E  0D           7.0  T16RPT: MVI     A,@15           ;SET FOR SUBGROUP 15 TEST
3061  4687  32  D6  48       13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
3062  468A  CD  52  47       18.0         CALL    FINDIT          ;GET THE PORT NUMBER
3063  468D                               SETLPC  T16LOOP,T16CONT
 (1)  468D  21  9A  46       10.0                 LXI     H,T16LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4690  22  E4  48       16.0                 SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  4693  21  AB  46       10.0                 LXI     H,T16CONT               ;GET THE CONTINUE ADDRESS
 (1)  4696  22  E7  48       16.0                 SHLD    ERCADR+1        ;SAVE IT
 (1)  4699  AF                4.0                 XRA     A
3064
3065  469A  CD  2F  48       18.0  T16LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
3066  469D  CA  A5  46       10.0         JZ      T16DE0          ;JUMP IF NO ERROR DETECTED
3067  46A0                         ERRB    XDLOOP,T16DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46A0  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000E                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A3  0E                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46A4  04                                 .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  46A5  CD  15  28      18.0         T16DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46A8  DA  89  48      10.0                 JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
3068                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
3069
3070  46AB  CD  A4  48       18.0  T16CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
3071  46AE  C2  9A  46       10.0         JNZ     T16LOOP         ;LOOP TILL 5 BYTES CHECKED
3072  46B1                         ENDTST  T16RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46B1                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46B1  CD  06  28      18.0                 CALL    REQST
 (2)  46B4  00                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  46B5  00  00                             .WORD                   ;SYSTEM '*' COUNT
 (2)  46B7  00  00                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46B9  00                                 .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46BA  07                                 .BYTE   7               ;REQUEST CODE
 (1)  46BB  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  46BE  3D               4.0         DCR     A               ;DOWNCOUNT
 (1)  46BF  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
 (1)  46C2  F2  85  46      10.0         JP      T16RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 17 - XMC - GCR 4X5 TRANSLATION SUBGROUP 16 TEST
                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1110(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01110(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
3119                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
3120                                    ;*      :       :       ENDDO
3121                                    ;*      :       :       IF TIMEOUT COUNT = 0
3122                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
3123                                    ;*      :       :       :       ELSE - CONTINUE
3124                                    ;*      :       :       ENDIF
3125                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
3126                                    ;*      :       ENDDO
3127                                    ;*      :       BGNDO
3128                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
3129                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
3130                                    ;*      :       :       :       THEN - CONTINUE
3131                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
3132                                    ;*      :       :       ENDIF
3133                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
3134                                    ;*      :       :       BGNDO
3135                                    ;*      :       :       :       CLOCK THE SYSTEM
3136                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
3137                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
3138                                    ;*      :       :       ENDDO
3139                                    ;*      :       :       IF 'WCLK' IS SET
3140                                    ;*      :       :       :       THEN - CONTINUE
3141                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
3142                                    ;*      :       :       ENDIF
3143                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
3144                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
3145                                    ;*      :       ENDDO
3146                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
3147                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
3148                                    ;*      ENDDO
3149                                    ;*ENDTST
3150  46C5                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
3151                                    ;*XMC5 MICRO TEST 17
3152                                    ;*XMC5 MICRO ERROR 17
3153                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
3154                                    ;*M8958, M8959, M8955'S
3155                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
3156                                    ;*ACTUAL = NNNN
3157                                    ;*EXPECTED = NNNN
3158                                    ;*TRANSITION COUNT = LLL
3159                                    ;*
3160                                    ;*XMC5 MICRO TEST 17
3161                                    ;*XMC5 MICRO ERROR 21
3162                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
3163                                    ;*M8958, M8959, M8955'S
3164                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
3165                                    ;*FATAL ERROR - TEST ABORTED
3166                                    ;*
3167                                    ;*XMC5 MICRO TEST 17
3168                                    ;*XMC5 MICRO ERROR 22
3169                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 17
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
3179  46C5                              S
 (1)                                    ; ***************************************************************************
3180  46C5                              TEST17: TESTX   @17
 (1)  46C5  3E  0F           7.0                  MVI     A,@17           ;DEFINE THE TEST NUMBER
 (1)  46C7  CD  03  28      18.0                  CALL    TSET            ;SETUP THE TEST
3181                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1110(2)
3182                                    ;&M8958, M8959, M8955'S
3183
3184  46CA  3E  0E           7.0  T17RPT: MVI     A,@16           ;SET FOR SUBGROUP 16 TEST
3185  46CC  32  D6  48      13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
3186  46CF  CD  52  47      18.0          CALL    FINDIT          ;GET THE PORT NUMBER
3187  46D2                                SETLPC  T17LOOP,T17CONT
 (1)  46D2  21  DF  46      10.0                  LXI     H,T17LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  46D5  22  E4  48      16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  46D8  21  F0  46      10.0                  LXI     H,T17CONT               ;GET THE CONTINUE ADDRESS
 (1)  46DB  22  E7  48      16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1)  46DE  AF               4.0                  XRA     A
3188
3189  46DF  CD  2F  48      18.0  T17LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
3190  46E2  CA  EA  46      10.0          JZ      T17DE0          ;JUMP IF NO ERROR DETECTED
3191  46E5                                ERRB    XDLOOP,T17DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46E5  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E8  0F                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46E9  04                                  .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  46EA  CD  15  28      18.0  T17DE0::              CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46ED  DA  89  48      10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
3192                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
3193
3194  46F0  CD  A4  48      18.0  T17CONT: CALL   NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
3195  46F3  C2  DF  46      10.0          JNZ     T17LOOP         ;LOOP TILL 5 BYTES CHECKED
3196  46F6                                ENDTST  T17RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46F6                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46F6  CD  06  28      18.0                  CALL    REQST
 (2)  46F9  00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  46FA  00  00                              .WORD                   ;SYSTEM "#" COUNT
 (2)  46FC  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46FE  00                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46FF  07                                  .BYTE   7               ;REQUEST CODE
 (1)  4700  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4703  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4704  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  4707  F2  CA  46      10.0          JP      T17RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 20 - XMC - GCR 4X5 TRANSLATION SUBGROUP 17 TEST
3199  470A                     ST
                               ;*********************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*XMC - GCR 4 X 5 TEST
3201  470A                     SD
                               ;*********************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                               ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                               ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                               ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1111(2) TO TEST TRANSLATION ROM
                               ;*PATTERN 01111(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                               ;*
                               ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                               ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                               ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                               ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
3212  470A                     SP
                               ;*********************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                               ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                               ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                               ;*      :       ELSE - CONTINUE
                               ;*      ENDIF
                               ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                               ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                               ;*      BGNDO
                               ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                               ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                               ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                               ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                               ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                               ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                               ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                               ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                               ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                               ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                               ;*      :       BGNDO
                               ;*      :       :       CLOCK THE SYSTEM
                               ;*      :       :       DECREMENT THE LOOP COUNT
                               ;*      :       :       DO UNTIL LOOP COUNTER = 0
                               ;*      :       ENDDO
                               ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                               ;*      :       BGNDO
                               ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                               ;*      :       :       BGNDO
                               ;*      :       :       :       CLOCK THE SYSTEM
                               ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
3243                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
3244                                    ;*      :       :       ENDDO
3245                                    ;*      :       :       IF TIMEOUT COUNT = 0
3246                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
3247                                    ;*      :       :       :       ELSE - CONTINUE
3248                                    ;*      :       :       ENDIF
3249                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
3250                                    ;*      :       ENDDO
3251                                    ;*      :       BGNDO
3252                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
3253                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
3254                                    ;*      :       :       :       THEN - CONTINUE
3255                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
3256                                    ;*      :       :       ENDIF
3257                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
3258                                    ;*      :       :       BGNDO
3259                                    ;*      :       :       :       CLOCK THE SYSTEM
3260                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
3261                                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
3262                                    ;*      :       :       ENDDO
3263                                    ;*      :       :       IF 'WCLK' IS SET
3264                                    ;*      :       :       :       THEN - CONTINUE
3265                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
3266                                    ;*      :       :       ENDIF
3267                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
3268                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
3269                                    ;*      :       ENDDO
3270                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
3271                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
3272                                    ;*      ENDDO
3273                                    ;*ENDTST
3274  470A                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
3275                                    ;*XMC5 MICRO TEST 20
3276                                    ;*XMC5 MICRO ERROR 20
3277                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
3278                                    ;*M8958, M8959, M8955'S
3279                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
3280                                    ;*ACTUAL = NNNN
3281                                    ;*EXPECTED = NNNN
3282                                    ;*TRANSITION COUNT = LLL
3283                                    ;*
3284                                    ;*XMC5 MICRO TEST 20
3285                                    ;*XMC5 MICRO ERROR 21
3286                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
3287                                    ;*M8958, M8959, M8955'S
3288                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
3289                                    ;*FATAL ERROR - TEST ABORTED
3290                                    ;*
3291                                    ;*XMC5 MICRO TEST 20
3292                                    ;*XMC5 MICRO ERROR 22
3293                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 20
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
                                        $
                                        ; ***************************************************************************
                                        TEST20: TESTX   @20
                                                        MVI     A,@20           ;DEFINE THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1111(2)
                                        ;&M8958, M8959, M8955'S
                                        T20RPT: MVI     A,@17           ;SET FOR SUBGROUP 17 TEST
                                                STA     XFRNUM          ;SAVE THE XFR NUMBER
                                                CALL    FINDIT          ;GET THE PORT NUMBER
                                                SETLPC  T20LOOP,T20CONT
                                                        LXI     H,T20LOOP               ;GET THE LOOP-ON-ERR ADDRESS
                                                        SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
                                                        LXI     H,T20CONT               ;GET THE CONTINUE ADDRESS
                                                        SHLD    ERCADR+1        ;SAVE IT
                                                        XRA     A

                                        T20LOOP: CALL   XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
                                                JZ      T20DE0          ;JUMP IF NO ERROR DETECTED
                                                ERRB    XDLOOP,T20DE0,4
                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE   4               ;PRINT ROUTINE NUMBER
                                        T20DE0::                CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
                                        T20CONT: CALL   NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
                                                JNZ     T20LOOP         ;LOOP TILL 5 BYTES CHECKED
                                                ENDTST  T20RPT
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
                                                        CALL    REQST
                                                        .BYTE                   ;DATA PATTERN NUMBER
                                                        .WORD                   ;SYSTEM '#' COUNT
                                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                        .BYTE                   ;DATA COMPARE FLAG IF =1
                                                        .BYTE   7               ;REQUEST CODE
                                                LDA     ITERA           ;GET ITERATION COUNT
                                                DCR     A               ;DOWNCOUNT
                                                STA     ITERA           ;SAVE COUNT
                                                JP      T20RPT          ;DO TEST UNTIL TILL = 0
                                                JMP     TSTEND          ;ALL DONE ... EXIT
```

```
3321                                    .SBTTL  FINDIT - SUBROUTINE TO GET A TEST PORT FOR TESTING USE
3322                                    ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
3323
3324  4752                              FINDIT: REQ     7
 (1)  4752  CD  06  28        18.0                 CALL    REQST
 (1)  4755  00                                     .BYTE                   ;DATA PATTERN NUMBER
 (1)  4756  00  00                                 .WORD                   ;SYSTEM '*' COUNT
 (1)  4758  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  475A  00                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  475B  07                                     .BYTE   7               ;REQUEST CODE
3325  475C                                    RIN     R12L                ;GET THE SELECTED PORT FROM 'HOST'
 (1)  475C  DB  94            10.0                 IN      R12L                ;READ R12L INTO AC
 (1)  475E  7F                 4.0                 MOV     A,A                 ;RETRY LINK
3326  475F  32  D7  48         13.0        STA     UNITMP              ;SAVE IT
3327  4762  A7                  4.0        ANA     A                   ;SET THE CONDITION BITS
3328  4763  C2  74  47         10.0        JNZ     FOUND               ;SEE IF ANY WERE SELECTED IN 'HOST'
3329  4766                               ERR     TSTEND,NOUNIT
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4766  CD  09  28        18.0                 CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0011                               MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4769  11                                 .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  476A  00                                 .BYTE
 (1)  476B  CD  15  28        18.0            NOUNIT::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  476E  DA  18  28        10.0                    JC      TSTEND              ;LOOP ADDRESS IF LOOP SPECIFIED
3330                                    ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
3331                                    ;>FATAL ERROR - TEST ABORTED
3332  4771  C3  18  28         10.0               JMP     TSTEND
3333                                    ;HERE IF THE HOST CPU INDICATED A PORT TO USE
3334
3335  4774  CD  22  4A         18.0   FOUND:  CALL    CLEAR               ;INIT ALL THE PORTS IN THE PORT BOARD
3336  4777  3A  D7  48         13.0           LDA     UNITMP              ;GET THE RESPONSE FROM 'HOST' CPU
3337  477A  06  00                7.0        MVI     B,0                 ;INIT 'B'
3338  477C  E6  01                7.0        ANI     1                   ;IS UNIT 0 SELECTED?
3339  477E  C2  94  47         10.0          JNZ     FPORT               ;YUP - FOUND THE PORT TO USE
3340  4781  04                   4.0        INR     B                   ;NO - COUNT THE PORT
3341  4782  3A  D7  48         13.0          LDA     UNITMP              ;GET RESPONSE AGAIN
3342  4785  E6  02                7.0        ANI     2                   ;PORT 1 SELECTED?
3343  4787  C2  94  47         10.0          JNZ     FPORT               ;YUP
3344  478A  04                   4.0        INR     B                   ;COUNT THE PORT
3345  478B  3A  D7  48         13.0          LDA     UNITMP              ;GET RESPONSE
3346  478E  E6  04                7.0        ANI     4                   ;PORT 2 SELECTED?
3347  4790  C2  94  47         10.0          JNZ     FPORT               ;YUP
3348  4793  04                   4.0        INR     B                   ;NO - ASSUME PORT 3
3349
3350  4794                           FPORT:
3351  4794  DB  E0                10.0        IN      INTSTA              ;GET THE PORT SELECT STATUS
3352  4796  E6  80                 7.0        ANI     BIT7
3353  4798  B0                     4.0        ORA     B                   ;ADD IN THE SELECTED PORT #
3354  4799  D3  E0                10.0        OUT     MBSEL               ;SELECT THE DESIRED PORT TO USE
3355  479B  3E  40                 7.0        MVI     A,P.LCS             ;GET THE LOOP-COMMAND-STATUS BIT
3356  479D  D3  48                10.0        OUT     PDIAG               ;SELECT IT
3357  479F  AF                     4.0        XRA     A
3358  47A0  D3  40                10.0        OUT     TCMD                ;CLEAR PORT COMMAND
3359  47A2  3E  10                 7.0        MVI     A,P.WPEN            ;SET WRITE PATH ENABLE
```

```
3360  47A4  D3  4C            10.0          OUT     PENAB           ;NOW
3361
3362                                      ;INIT THE COUNTER CONTROL AND COUNTERS
3363
3364  47A6                                INICNT
 (1)  47A6  3E  30             7.0                  MVI     A,@060          ;INIT THE
 (1)  47A8  D3  D7            10.0                  OUT     CNTCTL          ;BYTE COUNTER
 (1)  47AA  3E  70             7.0                  MVI     A,@160          ;INIT THE
 (1)  47AC  D3  D7            10.0                  OUT     CNTCTL          ;PAD COUNTER
 (1)  47AE  3E  B0             7.0                  MVI     A,@260          ;INIT THE
 (1)  47B0  D3  D7            10.0                  OUT     CNTCTL          ;ERROR COUNTER
3365  47B2                                CLRECT
 (1)  47B2  AF                 4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  47B3  D3  D6            10.0                  OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  47B5  D3  D6            10.0                  OUT     ERRCNT          ;CLEAR BITS 15-8
3366
3367                                      ;INIT THE SUPPLY TABLE INDEX POINTER
3368  47B7  3E  FF             7.0          MVI     A,@377
3369  47B9  32  E1  48        13.0          STA     DPNUM
3370  47BC  AF                  4.0          XRA     A
3371  47BD  32  DF  48        13.0          STA     CLKCNT          ;INIT THE 'SCLK' COUNTER
3372  47C0  32  D3  48        13.0          STA     TRANSCT         ;INIT THE TRANSITION COUNTER
3373  47C3                                  ROUT    R05H            ;CLEAR HIGH BYTE OF TRANS CNT REG (CAS)
 (1)  47C3  D3  8B            10.0                  OUT     R05H            ;WRITE AC INTO R05H
 (1)  47C5  7F                 4.0                  MOV     A,A             ;RETRY LINK
3374                                      ;INIT POINTERS FOR TEST PASS
3375
3376  47C6  3A  D6  48        13.0          LDA     XFRNUM          ;GET THE SUBGROUP NUMBER (0 - 17)
3377  47C9  17                  4.0          RAL                     ;MULT BY 4 - 2 LOCATIONS PER ENTRY
3378  47CA  17                  4.0          RAL
3379  47CB  E6  FC              7.0          ANI     @374            ;STRIP OFF THE TWO LOW BITS
3380  47CD  4F                  4.0          MOV     C,A             ;SAVE OFFSET INTO THE TABLE
3381  47CE  06  00              7.0          MVI     B,0
3382  47D0  21  E9  48        10.0          LXI     H,XFRTBL        ;POINT TO THE TABLE - BASIC POINTER
3383  47D3  09                 10.0          DAD     B               ;ADD IN THE OFFSET VALUE TO TABLE
3384  47D4  5E                  7.0          MOV     E,M             ;GET THE 'SUPPLY' DATA ADDRESS LOW BYTE
3385  47D5  23                  6.0          INX     H               ;POINT TO THE HIGH BYTE OF ADDRESS
3386  47D6  56                  7.0          MOV     D,M             ;GET THE HIGH BYTE OF ADDRESS
3387  47D7  EB                  4.0          XCHG
3388  47D8  22  D8  48        16.0          SHLD    SUPADR          ;SAVE THE ADDRESS OF SUPPLY DATA TABLE
3389  47DB  EB                  4.0          XCHG
3390  47DC  23                  6.0          INX     H               ;POINT TO THE EXPECTED DATA TABLE ADDRESS
3391  47DD  5E                  7.0          MOV     E,M             ;GET LOW BYTE OF ADDRESS
3392  47DE  23                  6.0          INX     H               ;POINT TO THE HIGH BYTE OF ADDRESS
3393  47DF  56                  7.0          MOV     D,M             ;GET THE HIGH BYTE OF ADDRESS
3394  47E0  EB                  4.0          XCHG
3395  47E1  22  DA  48        16.0          SHLD    EXPADR          ;SAVE THE ADDRESS OF THE EXPECTED DATA TABLE
3396  47E4  EB                  4.0          XCHG
3397  47E5  CD  FD  49        18.0          CALL    GENDAT          ;LOAD 1ST SET OF DATA FOR XFR
3398
3399                                      ;ISSUE A SYSTEM 'RESTART'
3400
3401  47E8  3E  01              7.0  TSTRST: MVI     A,W.RST         ;GET THE RESTART BIT
3402  47EA  D3  D3             10.0          OUT     WMCCTL
```

```
3403  47EC  3E  05        7.0        MVI   A,SSTEP          ;GET SINGLE-STEP
3404  47EE  D3  F0       10.0        OUT   CLKCTL           ;SELECT IT
3405  47F0  32  D5  48   13.0        STA   CCTLWD           ;SAVE CLOCK CONTROL WORD
3406  47F3  CD  16  4A   18.0        CALL  CLKSYS           ;CLOCK THE SYSTEM TO MAKE THE RESTART
3407  47F6  CD  16  4A   18.0        CALL  CLKSYS           ;TAKE EFFECT
3408  47F9  AF             4.0        XRA   A
3409  47FA  D3  D3        10.0        OUT   WMCCTL           ;FINISH THE RESTART CYCLE
3410
3411                               ;LOAD PROPER BYTE COUNT & PAD COUNT FOR A 4 BYTE TRANSFER
3412
3413  47FC  3E  01         7.0        MVI   A,1
3414  47FE  D3  D4        10.0        OUT   BYTCNT           ;BYTE COUNTER LOADED WITH A 1 TO
3415  4800  AF             4.0        XRA   A                ;INDICATE NO DATA GROUP TO BE ISSUED
3416  4801  D3  D4        10.0        OUT   BYTCNT           ;JUST A RESIDUAL DATA GROUP
3417  4803  3E  03         7.0        MVI   A,3              ;PAD COUNTER FOR 4 BYTE DATA XFR = 3
3418  4805  D3  D5        10.0        OUT   PADCNT           ;2 PADS PLUS AN EXTRA COUNT TO SATISIFY
3419  4807  AF             4.0        XRA   A                ;THE INTEL COUNTER CHIP
3420  4808  D3  D5        10.0        OUT   PADCNT
3421                               ;INIT THE 'DDR' CONTROL TO 'OUT' SO DATA CAN BE INJECTED INTO THE WMC
3422
3423  480A  3E  88         7.0        MVI   A,@210
3424  480C  D3  DB        10.0        OUT   DDRCTL           ;SET DDR TO 'OUT'
3425
3426                               ;SELECT DATA FORMAT FOR THE XFR - 'IMAGE' MODE
3427
3428  480E  3E  D0         7.0        MVI   A,$50!BIT7       ;IMAGE MODE - 1 BYTE EVERY 'SCLK' TIME
3429  4810  D3  D0        10.0        OUT   DATACTL
3430
3431                               ;SETUP THE RESIDUAL GROUP CONTROL WORD
3432
3433  4812  3E  83         7.0        MVI   A,@203           ;SET TO INDICATE 4 BYTES OF DATA IN RESID
3434  4814  D3  D1        10.0        OUT   RESCHR           ;DATA GROUP
3435
3436                               ;ENABLE ALL THE DATA TRACKS
3437
3438  4816  AF             4.0        XRA   A
3439  4817  D3  D2        10.0        OUT   TRKENA           ;PRESET ALL TRACKS TO 0 SO THE
3440  4819  D3  D2        10.0        OUT   TRKENA           ;PREAMBLE WILL HAVE A PREDICTED RESULT
3441  481B  3E  FF         7.0        MVI   A,@377
3442  481D  D3  D2        10.0        OUT   TRKENA           ;ENABLE TRACKS 7-0
3443  481F  D3  D2        10.0        OUT   TRKENA           ;ENABLE 'P' TRACK
3444
3445                               ;CLOCK THROUGH THE WMC RESTART AREA
3446
3447  4821  3E  B4         7.0        MVI   A,180            ;SETUP A LOOP COUNT
3448  4823  CD  16  4A    18.0  CLKRS: CALL  CLKSYS          ;CLOCK THE SYSTEM
3449  4826  05             4.0        DCR   B                ;DECREMENT THE LOOP COUNTER
3450  4827  C2  23  48    10.0        JNZ   CLKRS            ;CLOCK TILL COUNTER =0
3451
3452                               ;START THE DATA TRANSFER - GCR DATA WRITE XFR
3453
3454  482A  3E  D8         7.0        MVI   A,W.ENAB.X.ENAB!W.GCR!W.WRITE
3455  482C  D3  D3        10.0        OUT   WMCCTL           ;START THE XFR NOW
3456  482E  C9            10.0        RET                    ;EXIT THE 'FINDIT' SUBROUTINE
```

```
                                        .SBTTL  XFRSUB - SUBROUTINE TO CLOCK DATA THRU TRANSLATOR & CHECK IT
                                        ;'WATCHDOG' LOOP
                                        
482F  FA 5B 48        10.0  XFRSUB: JM      XMCDAT          ;JUMP IF HERE BEFORE (CALLED FROM NXTBYT)
4832  06 FF            7.0  CLK1:   MVI     B,@377
                                        
4834  CD B9 49        18.0  CLK:    CALL    CKSCLK          ;CLOCK SYSTEM & CHECK FOR SCLK
4837  DA 56 48        10.0          JC      FEXIT           ;JUMP IF ERROR DETECTED
483A  DB D0             10.0          IN      WMCSTA          ;GET THE WMC STATUS
483C  32 E2 48          13.0          STA     SWMCSTA         ;SAVE THIS STATUS
483F  E6 01              7.0          ANI     X.WCLK          ;TRANSLATOR WRITE CLOCK SET?
4841  C2 5B 48        10.0          JNZ     XMCDAT          ;YES - JUMP IF READY
4844  05                4.0          DCR     B               ;NO - DECREMENT THE LOOP COUNTER
4845  C2 34 48        10.0          JNZ     CLK             ;JUMP BACK IF LOOP COUNTER NOT =0
4848                                  ERR     TSTEND,CLKER0
 (1)                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4848  CD 09 28     18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0012                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  484B  12                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  484C  00                        .BYTE
 (1)  484D  CD 15 28     18.0  CLKER0::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4850  DA 18 28     10.0                 JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
4853  C3 18 28        10.0          JMP     TSTEND
                                        
4856  33                6.0  FEXIT:  INX     SP              ;CORRECT THE STACK SO FORCE EXIT WILL
4857  33                6.0          INX     SP              ;WORK CORRECTLY
4858  C3 E3 48        10.0          JMP     LPCADR          ;TAKE THE LOOP-ON-ERROR EXIT
                                        ;HERE TO CHECK THE OCCURANCE OF XMC 'WCLK'
                                        
485B  CD B9 49        18.0  XMCDAT: CALL    CKSCLK          ;CLOCK THE SYSTEM TWO MORE TIMES
485E  CD B9 49        18.0          CALL    CKSCLK          ;TO 'LOCK' THE DATA IN THE PORT BOARD
4861  DB 40           10.0          IN      TSTS            ;GET THE ACTUAL DATA FROM PORT BOARD
4863  47               4.0          MOV     B,A             ;TEMP SAVE IT
4864                               ROUT    ADATA           ;SAVE IN 'CAS' FOR ERROR REPORTING
 (1)  4864  D3 94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4866  7F            4.0                 MOV     A,A             ;RETRY LINK
                                        
                                        ;SEE IF THIS TRANSITION IS TO BE CHECKED - MUST BE PAST THE GCR
                                        ;PREAMBLE AND THE MARK 1 BURST & END MARK BURST BUT NOT INTO THE
                                        ;2ND SUBGROUP OF DATA
                                        
4867  3A D3 48        13.0          LDA     TRANSCT         ;GET THE CURRENT TRANSITION COUNT
486A  3C               4.0          INR     A               ;COUNT THIS OCCURANCE OF 'WCLK'
486B  32 D3 48          13.0          STA     TRANSCT         ;SAVE THE UPDATED COUNT
486E                               ROUT    R05L            ;SAVE THE COUNT IN CAS FOR ERROR REPORT
 (1)  486E  D3 8A        10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  4870  7F            4.0                 MOV     A,A             ;RETRY LINK
4871  FE 5B             7.0          CPI     @133            ;INTO THE DATA SUBGROUP YET?
4873  DA 32 48        10.0          JC      CLK1            ;NO - KEEP CLOCKING THE SYSTEM
                                        
                                        ;CHECK THE DATA FROM THE 'EXPECTED' DATA TABLE
                                        
4876  2A DA 48        16.0          LHLD    EXPADR          ;GET THE ADDRESS OF THE DATA TABLE
```

```
3501  4879  7E              7.0          MOV     A,M             ;GET THE EXPECTED DATA BYTE
3502  487A                               ROUT    EDATA           ;SAVE IT IN THE 'CAS' FOR ERROR REPORTING
 (1)  487A  D3  95           10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  487C  7F               4.0                  MOV     A,A             ;RETRY LINK
3503  487D  B8               4.0          CMP     B               ;ACTUAL = EXPECTED DATA??
3504  487E  C8               12.0         RZ                      ;TAKE THE 'OK' EXIT IF NO COMPARE ERROR
3505  487F  3A  D3  48        13.0         LDA     TRANSCT         ;ERROR DETECTED - GET THE TRANSITION COUNT
3506  4882  32  D4  48        13.0         STA     EXDLPC          ;SAVE IT FOR THE ERROR LOOPING CONTROL
3507  4885  3E  FF            7.0          MVI     A,$FF           ;SET A 'FLAG' FOR ERROR EXIT
3508  4887  A7               4.0          ANA     A
3509  4888  C9               10.0         RET                     ;ERROR EXIT
3510                                   ;HERE IS THE ERROR LOOP CONTROL
3511
3512  4889  3A  94  4F        13.0 XDLOOP: LDA     LPFLG           ;GET LOOP-ON-ERROR FLAG
3513  488C  A7               4.0          ANA     A
3514  488D  C2  96  48        10.0         JNZ     XDLP1           ;JUMP IF LOOP FLAG IS SET
3515  4890  32  D4  48        13.0         STA     EXDLPC          ;NO - CLEAR THE LOOP CONTROL
3516  4893  C3  E6  48        10.0         JMP     ERCADR          ;TAKE THE PROCEED EXIT
3517
3518  4896  3A  D3  48        13.0 XDLP1:  LDA     TRANSCT         ;GET THE CURRENT TRNASITION COUNT
3519  4899  47               4.0          MOV     B,A             ;TEMP SAVE
3520  489A  3A  D4  48        13.0         LDA     EXDLPC          ;GET THE LOOP CONTROL COUNT
3521  489D  B8               4.0          CMP     B               ;LOOP AT THIS TRANSITION COUNT?
3522  489E  CA  E3  48        10.0         JZ      LPCADR          ;TAKE THE ERROR LOOP ADDRESS EXIT
3523  48A1  C3  E6  48        10.0         JMP     ERCADR          ;NO MATCH...TAKE THE PROCEED EXIT
3524
3525                                   ;UPDATE THE DATA POINTER FOR THE NEXT 'WCLK' SIGNAL
3526
3527  48A4  2A  DA  48        16.0 NXTBYT: LHLD    EXPADR          ;GET THE EXPECTED DATA TABLE ADDRESS
3528  48A7  23               6.0          INX     H               ;UPDATE THE POINTER
3529  48A8  22  DA  48        16.0         SHLD    EXPADR          ;SAVE THE POINTER FOR NEXT PASS
3530  48AB  3A  D3  48        13.0         LDA     TRANSCT         ;GET THE CURRENT TRANSITION COUNT
3531  48AE  FE  5F            7.0          CPI     @137            ;IS THIS THE LAST BYTE IN  SUBGROUP?
3532  48B0  C8               12.0         RZ                      ;EXIT - 'Z' SET IF ALL DONE
3533
3534                                   ;CLOCK THE SYSTEM 14(8) MORE TIMES TO GET NEXT BYTE
3535
3536  48B1  06  0E            7.0  CLKNXT: MVI     B,14            ;SETUP THE LOOP COUNT
3537  48B3  CD  B9  49        18.0 NXTB1:  CALL    CKSCLK          ;CLOCK THE SYSTEM
3538  48B6  05               4.0          DCR     B               ;DECREMENT THE LOOP COUNT
3539  48B7  C2  B3  48        10.0         JNZ     NXTB1           ;CLOCK TILL THE COUNT =0
3540
3541                                   ;CHECK TO SEE IF 'WCLK' IS SET AFTER 14 CLOCKS - NEXT BYTE READY
3542
3543  48BA  DB  D0            10.0         IN      WMCSTA          ;GET THE WMC STATUS
3544  48BC  32  E2  48        13.0         STA     SWMCSTA         ;SAVE THIS STATUS
3545  48BF  E6  01            7.0          ANI     X.WCLK          ;IS THE WCLK BIT SET?
3546  48C1  C2  C9  48        10.0         JNZ     NOCLKC          ;YUP - OK TO PROCEED
```

```
3548  48C4                                          ERR     CLK1,NOCLKC,4
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  48C4  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0013                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48C7  13                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48C8  04                                             .BYTE   4
 (1)  48C9  CD  15  28        18.0                  NOCLKC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  48CC  DA  32  48        10.0                          JC      CLK1            ;LOOP ADDRESS IF LOOP SPECIFIED
3549                                               ;>TRANSLATOR WRITE CLOCK NOT SET AFTER 8 CLOCK CYCLES
3550                                               ;>IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
3551                                               ;*XMC5 MICRO ERROR 23
3552                                               ;*TRANSLATOR WRITE CLOCK NOT SET AFTER 8 CLOCK CYCLES
3553                                               ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
3554                                               ;*TRANSITION COUNT = LLL
3555  48CF  3E  FF                7.0               MVI     A,$FF           ;SET A FLAG
3556  48D1  A7                    4.0               ANA     A               ;AND THE CONDITION BITS
3557  48D2  C9                   10.0               RET                     ;EXIT
3558                                             ;HERE IS THE VARIABLE DATA STORAGE AREA AND THE FLAG STORAGE AREA
3559
3560  48D3  00                                   TRANSCT:        .BYTE   0       ;TRANSITION COUNTER
3561  48D4  00                                   EXDLPC:         .BYTE   0       ;TRANSITION COUNT AT DETECTED ERROR
3562  48D5  00                                   CCTLWD:         .BYTE   0       ;CLOCK CONTROL WORD - COPY
3563  48D6  00                                   XFRNUM:         .BYTE   0       ;4 X 5 SUBGROUP NUMBER
3564  48D7  00                                   UNITMP:         .BYTE   0       ;SELECTED PORT # FROM 'HOST' CPU
3565  48D8  00  00                               SUPADR:         .ADDR   0       ;ADDRESS OF THE 'SUPPLY' DATA TABLE
3566  48DA  00  00                               EXPADR:         .ADDR   0       ;ADDRESS OF THE 'EXPECTED' DATA TABLE
3567
3568  48DC  00                                   SDDRAT:         .BYTE   0       ;DATA TO DDR REG A
3569  48DD  00                                   SDDRA:          .BYTE   0       ;SHIFTED DDR REG A DATA (BITS 5-0,17-16)
3570  48DE  00                                   SDDRB:          .BYTE   0       ;   "    "   "  B   " (BITS 13-6)
3571
3572  48DF  00                                   CLKCNT:         .BYTE   0       ;COUNT OF 'SCLK' OCCURANCES
3573  48E0  00                                   CLKON:          .BYTE   0       ;SOFTWARE FLAG FOR 'SCLK' ON
3574  48E1  00                                   DPNUM:          .BYTE   0       ;DATA BYTE INDEX NUMBER FOR SUPPLY DATA
3575  48E2  00                                   SWMCSTA:        .BYTE   0       ;SOFTWARE COPY OF WMCSTA WORD
3576
3577                                             ;STORAGE FOR THE ERROR LOOP CONTROL
3578
3579  48E3  C3  18  28        10.0               LPCADR:         JMP     TSTEND  ;FAKE FATAL EXIT - NORMALLY SET BY TEST
3580  48E6  C3  18  28        10.0               ERCADR:         JMP     TSTEND
```

```
3582                                    ;POINTER TABLE FOR 'SUPPLY' AND 'EXPECTED' DATA TABLE ADDRESSES
3583
3584  48E9  29  49    XFRTBL: .ADDR   SUP00   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 00
3585  48EB  69  49            .ADDR   EXP00   ;EXPECTED DATA TABLE FOR SUBGROUP 00
3586  48ED  2D  49            .ADDR   SUP01   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 01
3587  48EF  6E  49            .ADDR   EXP01   ;EXPECTED DATA TABLE FOR SUBGROUP 01
3588  48F1  31  49            .ADDR   SUP02   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 02
3589  48F3  73  49            .ADDR   EXP02   ;EXPECTED DATA TABLE FOR SUBGROUP 02
3590  48F5  35  49            .ADDR   SUP03   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 03
3591  48F7  78  49            .ADDR   EXP03   ;EXPECTED DATA TABLE FOR SUBGROUP 03
3592  48F9  39  49            .ADDR   SUP04   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 04
3593  48FB  7D  49            .ADDR   EXP04   ;EXPECTED DATA TABLE FOR SUBGROUP 04
3594  48FD  3D  49            .ADDR   SUP05   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 05
3595  48FF  82  49            .ADDR   EXP05   ;EXPECTED DATA TABLE FOR SUBGROUP 05
3596  4901  41  49            .ADDR   SUP06   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 06
3597  4903  87  49            .ADDR   EXP06   ;EXPECTED DATA TABLE FOR SUBGROUP 06
3598  4905  45  49            .ADDR   SUP07   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 07
3599  4907  8C  49            .ADDR   EXP07   ;EXPECTED DATA TABLE FOR SUBGROUP 07
3600  4909  49  49            .ADDR   SUP10   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 10
3601  490B  91  49            .ADDR   EXP10   ;EXPECTED DATA TABLE FOR SUBGROUP 10
3602  490D  4D  49            .ADDR   SUP11   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 11
3603  490F  96  49            .ADDR   EXP11   ;EXPECTED DATA TABLE FOR SUBGROUP 11
3604  4911  51  49            .ADDR   SUP12   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 12
3605  4913  9B  49            .ADDR   EXP12   ;EXPECTED DATA TABLE FOR SUBGROUP 12
3606  4915  55  49            .ADDR   SUP13   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 13
3607  4917  A0  49            .ADDR   EXP13   ;EXPECTED DATA TABLE FOR SUBGROUP 13
3608  4919  59  49            .ADDR   SUP14   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 14
3609  491B  A5  49            .ADDR   EXP14   ;EXPECTED DATA TABLE FOR SUBGROUP 14
3610  491D  5D  49            .ADDR   SUP15   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 15
3611  491F  AA  49            .ADDR   EXP15   ;EXPECTED DATA TABLE FOR SUBGROUP 15
3612  4921  61  49            .ADDR   SUP16   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 16
3613  4923  AF  49            .ADDR   EXP16   ;EXPECTED DATA TABLE FOR SUBGROUP 16
3614  4925  65  49            .ADDR   SUP17   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 17
3615  4927  B4  49            .ADDR   EXP17   ;EXPECTED DATA TABLE FOR SUBGROUP 17
3616                          ;HERE IS THE 'SUPPLY' DATA TABLES FOR DATA INJECTION INTO THE WMC
3617                          ;WHEN 'SCLK' OCCURES
3618
3619                          ;SUBGROUP 00 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3620
3621  4929  00                SUP00:  .BYTE   0
3622  492A  00                        .BYTE   0
3623  492B  00                        .BYTE   0
3624  492C  00                        .BYTE   0
3625
3626                          ;SUBGROUP 01 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3627
3628  492D  00                SUP01:  .BYTE   0
3629  492E  00                        .BYTE   0
3630  492F  00                        .BYTE   0
3631  4930  FF                        .BYTE   $FF
3632
3633                          ;SUBGROUP 02 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3634
3635  4931  00                SUP02:  .BYTE   0
```

```
3636  4932  00                        .BYTE   0
3637  4933  FF                        .BYTE   $FF
3638  4934  00                        .BYTE   0
3639
3640                                  ;SUBGROUP 03 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3641
3642  4935  00                        SUP03:  .BYTE   0
3643  4936  00                                .BYTE   0
3644  4937  FF                                .BYTE   $FF
3645  4938  FF                                .BYTE   $FF
3646
3647                                  ;SUBGROUP 04 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3648
3649  4939  00                        SUP04:  .BYTE   0
3650  493A  FF                                .BYTE   $FF
3651  493B  00                                .BYTE   0
3652  493C  00                                .BYTE   0
3653
3654                                  ;SUBGROUP 05 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3655
3656  493D  00                        SUP05:  .BYTE   0
3657  493E  FF                                .BYTE   $FF
3658  493F  00                                .BYTE   0
3659  4940  FF                                .BYTE   $FF
3660                                  ;SUBGROUP 06 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3661
3662  4941  00                        SUP06:  .BYTE   0
3663  4942  FF                                .BYTE   $FF
3664  4943  FF                                .BYTE   $FF
3665  4944  00                                .BYTE   0
3666
3667                                  ;SUBGROUP 07 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3668
3669  4945  00                        SUP07:  .BYTE   0
3670  4946  FF                                .BYTE   $FF
3671  4947  FF                                .BYTE   $FF
3672  4948  FF                                .BYTE   $FF
3673
3674                                  ;SUBGROUP 10 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3675
3676  4949  FF                        SUP10:  .BYTE   $FF
3677  494A  00                                .BYTE   0
3678  494B  00                                .BYTE   0
3679  494C  00                                .BYTE   0
3680
3681                                  ;SUBGROUP 11 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3682
3683  494D  FF                        SUP11:  .BYTE   $FF
3684  494E  00                                .BYTE   0
3685  494F  00                                .BYTE   0
3686  4950  FF                                .BYTE   $FF
3687
3688                                  ;SUBGROUP 12 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3689
```

```
3690  4951  FF          SUP12:  .BYTE   $FF
3691  4952  00                  .BYTE   0
3692  4953  FF                  .BYTE   $FF
3693  4954  00                  .BYTE   0
3694
3695                    ;SUBGROUP 13 DATA FOR  .JECTION TO WMC VIA DDR REGISTERS
3696
3697  4955  FF          SUP13:  .BYTE   $FF
3698  4956  00                  .BYTE   0
3699  4957  FF                  .BYTE   $FF
3700  4958  FF                  .BYTE   $FF
3701
3702                    ;SUBGROUP 14 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3703
3704  4959  FF          SUP14:  .BYTE   $FF
3705  495A  FF                  .BYTE   $FF
3706  495B  00                  .BYTE   0
3707  495C  00                  .BYTE   0
3708                    ;SUBGROUP 15 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3709
3710  495D  FF          SUP15:  .BYTE   $FF
3711  495E  FF                  .BYTE   $FF
3712  495F  00                  .BYTE   0
3713  4960  FF                  .BYTE   $FF
3714
3715                    ;SUBGROUP 16 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3716
3717  4961  FF          SUP16:  .BYTE   $FF
3718  4962  FF                  .BYTE   $FF
3719  4963  FF                  .BYTE   $FF
3720  4964  00                  .BYTE   0
3721
3722                    ;SUBGROUP 17 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3723
3724  4965  FF          SUP17:  .BYTE   $FF
3725  4966  FF                  .BYTE   $FF
3726  4967  FF                  .BYTE   $FF
3727  4968  FF                  .BYTE   $FF
3728
3729  4969                    S
 (1)                    ; ***************************************************************************
3730                    ;HERE IS THE EXPECTED DATA TABLES FOR THE 4 X 5 SUBGROUP TRANSLATIONS
3731                    ;ALL EXPECTED DATA STARTS AT TRANSITION COUNT 133(8) AND ENDS AT 137(8)
3732  4969              S
 (1)                    ; ***************************************************************************
3733
3734                    ;EXPECTED SUBGROUP 00 DATA (11001 FOR ALL BITS)
3735
3736  4969  00          EXP00:  .BYTE   0
3737  496A  FF                  .BYTE   $FF
3738  496B  FF                  .BYTE   $FF
3739  496C  FF                  .BYTE   $FF
3740  496D  00                  .BYTE   0
3741
```

```
                                        ;EXPECTED SUBGROUP 01 DATA (11011 FOR ALL BITS)

496E  00                        EXP01:  .BYTE   0
496F  FF                                .BYTE   $FF
4970  FF                                .BYTE   $FF
4971  00                                .BYTE   0
4972  FF                                .BYTE   $FF

                                        ;EXPECTED SUBGROUP 02 DATA (10010 FOR ALL BITS)

4973  00                        EXP02:  .BYTE   0
4974  00                                .BYTE   0
4975  00                                .BYTE   0
4976  FF                                .BYTE   $FF
4977  FF                                .BYTE   $FF
                                        ;EXPECTED SUBGROUP 03 DATA (10011 FOR ALL BITS)

4978  00                        EXP03:  .BYTE   0
4979  00                                .BYTE   0
497A  00                                .BYTE   0
497B  FF                                .BYTE   $FF
497C  00                                .BYTE   0

                                        ;EXPECTED SUBGROUP 04 DATA (11101 FOR ALL BITS)

497D  00                        EXP04:  .BYTE   0
497E  FF                                .BYTE   $FF
497F  00                                .BYTE   0
4980  00                                .BYTE   0
4981  FF                                .BYTE   $FF

                                        ;EXPECTED SUBGROUP 05 DATA (10101 FOR ALL BITS)

4982  00                        EXP05:  .BYTE   0
4983  00                                .BYTE   0
4984  FF                                .BYTE   $FF
4985  FF                                .BYTE   $FF
4986  00                                .BYTE   0

                                        ;EXPECTED SUBGROUP 06 DATA (10110 FOR ALL BITS)

4987  00                        EXP06:  .BYTE   0
4988  00                                .BYTE   0
4989  FF                                .BYTE   $FF
498A  00                                .BYTE   0
498B  00                                .BYTE   0

                                        ;EXPECTED SUBGROUP 07 DATA (10111 FOR ALL BITS)

498C  00                        EXP07:  .BYTE   0
498D  00                                .BYTE   0
498E  FF                                .BYTE   $FF
498F  00                                .BYTE   0
4990  FF                                .BYTE   $FF
```

```
3796
3797                                    ;EXPECTED SUBGROUP 10 DATA (11010 FOR ALL BITS)
3798
3799  4991  00                          EXP10:  .BYTE   0
3800  4992  FF                                  .BYTE   $FF
3801  4993  FF                                  .BYTE   $FF
3802  4994  00                                  .BYTE   0
3803  4995  00                                  .BYTE   0
3804                                    ;EXPECTED SUBGROUP 11 DATA (01001 FOR ALL BITS)
3805
3806  4996  FF                          EXP11:  .BYTE   $FF
3807  4997  00                                  .BYTE   0
3808  4998  00                                  .BYTE   0
3809  4999  00                                  .BYTE   0
3810  499A  FF                                  .BYTE   $FF
3811
3812                                    ;EXPECTED SUBGROUP 12 DATA (01010 FOR ALL BITS)
3813
3814  499B  FF                          EXP12:  .BYTE   $FF
3815  499C  00                                  .BYTE   0
3816  499D  00                                  .BYTE   0
3817  499E  FF                                  .BYTE   $FF
3818  499F  FF                                  .BYTE   $FF
3819
3820                                    ;EXPECTED SUBGROUP 13 DATA (01011 FOR ALL BITS)
3821
3822  49A0  FF                          EXP13:  .BYTE   $FF
3823  49A1  00                                  .BYTE   0
3824  49A2  00                                  .BYTE   0
3825  49A3  FF                                  .BYTE   $FF
3826  49A4  00                                  .BYTE   0
3827
3828                                    ;EXPECTED SUBGROUP 14 DATA (11110 FOR ALL BITS)
3829
3830  49A5  00                          EXP14:  .BYTE   0
3831  49A6  FF                                  .BYTE   $FF
3832  49A7  00                                  .BYTE   0
3833  49A8  FF                                  .BYTE   $FF
3834  49A9  FF                                  .BYTE   $FF
3835
3836                                    ;EXPECTED SUBGROUP 15 DATA (01101 FOR ALL BITS)
3837
3838  49AA  FF                          EXP15:  .BYTE   $FF
3839  49AB  00                                  .BYTE   0
3840  49AC  FF                                  .BYTE   $FF
3841  49AD  FF                                  .BYTE   $FF
3842  49AE  00                                  .BYTE   0
3843                                    ;EXPECTED SUBGROUP 16 DATA (01110 FOR ALL BITS)
3844
3845  49AF  FF                          EXP16:  .BYTE   $FF
3846  49B0  00                                  .BYTE   0
3847  49B1  FF                                  .BYTE   $FF
3848  49B2  00                                  .BYTE   0
3849  49B3  00                                  .BYTE   0
```

```
                                        ;EXPECTED SUBGROUP 17 DATA (01111 FOR ALL BITS)

49B4  FF                                EXP17:  .BYTE   $FF
49B5  00                                        .BYTE   0
49B6  FF                                        .BYTE   $FF
49B7  00                                        .BYTE   0
49B8  FF                                        .BYTE   $FF
49B9                                    S
 (1)                                    ; ***************************************************************************
                                        ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
                                        ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
                                        ;       THE DDR.
                                        ;
49B9                                    S
 (1)                                    ; ***************************************************************************

49B9  CD  16  4A        18.0    CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
49BC  DB  C0            10.0            IN      DBUSSTA         ;GET STATUS OF DATA BUS
49BE  E6  02             7.0            ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
49C0  C2  C8  49        10.0            JNZ     SCLKBIT         ;JUMP IF SCLK IS ON
49C3  AF                 4.0            XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
49C4  32  E0  48        13.0            STA     CLKON
49C7  C9                10.0            RET                     ;EXIT

49C8                                    S
 (1)                                    ; ***************************************************************************
                                        ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
                                        ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
                                        ;
49C8                                    S
 (1)                                    ; ***************************************************************************

49C8  3A  E0  48        13.0    SCLKBIT: LDA    CLKON           ;SEE IF 'SCLK' WAS ON
49CB  A7                 4.0            ANA     A               ;SET CONDITION BITS
49CC  C0                12.0            RNZ                     ;EXIT IF CLOCK WAS SET - HAVE DATA ALREADY

49CD  CD  E1  49        18.0            CALL    SETDDR          ;LOAD THE DATA INTO THE DDR REGS
49D0  CD  FD  49        18.0            CALL    GENDAT          ;AND GENERATE NEW DATA FOR NEXT DDR LOAD

49D3  06  7F             7.0    SCKB1:  MVI     B,@177          ;GET A NEW WATCHDOG COUNT
49D5  78                 4.0            MOV     A,B             ;COPY IT TO REG A
49D6  32  E0  48        13.0            STA     CLKON           ;SET THE SCLK ON FLAG
49D9  3A  DF  48        13.0            LDA     CLKCNT          ;GET THE CURRENT SCLK COUNT
49DC  3C                 4.0            INR     A               ;UPDATE IT
49DD  32  DF  48        13.0            STA     CLKCNT          ;SAVE THE NEW COUNT
49E0  C9                10.0            RET                     ;EXIT
49E1                                    S
 (1)                                    ; ***************************************************************************
                                        ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
                                        ;       POSITION TO THE DDR TO BE USED AS DATA TO THE TM78.
                                        ;
                                        ;       THE DATA HAS TO BE SHIFTED TO COMPENSATE FOR THE 'LEFT' FLAG
                                        ;       CONTROL IN THE DDR REGISTERS. THE PACKING MODE USED IS 'IMAGE'.
                                        ;
49E1                                    S
 (1)                                    ; ***************************************************************************
```

```
3898
3899                                        ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR INJECTION
3900
3901  49E1  3A DC 48        13.0  SETDDR: LDA     SDDRAT          ;GET DDRA TEMP
3902  49E4  E6 3F            7.0          ANI     @77             ;STRIP 2 BITS OFF
3903  49E6  07                4.0          RLC
3904  49E7  07                4.0          RLC                     ;LEFT JUSTIFY
3905  49E8  32 DD 48         13.0          STA     SDDRA           ;SAVE DDRA DESIRED
3906  49EB  D3 D8            10.0          OUT     DDRA            ;LOAD THE REAL DDRA
3907
3908  49ED  3A DC 48         13.0          LDA     SDDRAT          ;GET DDRA AGAIN
3909  49F0  E6 C0             7.0          ANI     @300            ;SAVE 2 BITS
3910  49F2  07                4.0          RLC
3911  49F3  07                4.0          RLC
3912  49F4  32 DE 48         13.0          STA     SDDRB           ;SAVE DDRB DATA
3913  49F7  D3 D9            10.0          OUT     DDRB            ;LOAD THE REAL DDRB
3914
3915  49F9  AF                4.0          XRA     A
3916  49FA  D3 DA            10.0          OUT     DDRC
3917  49FC  C9               10.0          RET                     ;EXIT
3918  49FD                                S
 (1)                                        ; *****************************************************************************
3919                                        ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT ''SCLK'' CYCLE.
3920                                        ;       DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAT'
3921                                        ;       THRU 'SDDRCT' FOR REFERENCE.
3922  49FD                                S
 (1)                                        ; *****************************************************************************
3923
3924  49FD  F5               12.0  GENDAT: PUSH    PSW             ;SAVE PSW + A
3925  49FE  C5               12.0          PUSH    B               ;SAVE B + C
3926  49FF  E5               12.0          PUSH    H               ;SAVE H + L
3927  4A00  3A E1 48         13.0          LDA     DPNUM           ;GET THE LAST PATTERN # USED
3928  4A03  3C                4.0          INR     A               ;POINT TO NEXT
3929  4A04  32 E1 48         13.0          STA     DPNUM           ;SAVE NUMBER
3930  4A07  4F                4.0          MOV     C,A
3931  4A08  06 00             7.0          MVI     B,0             ;B + C HAVE OFFSET
3932  4A0A  2A D8 48         16.0          LHLD    SUPADR          ;GET THE BASE ADDRESS OF THE SUPPLY DATA
3933  4A0D  09               10.0          DAD     B               ;ADD IN OFFSET TO H + L
3934
3935  4A0E  7E                7.0          MOV     A,M             ;GET DATA BYTE FOR DDR 'A' (BITS 7-0)
3936  4A0F  32 DC 48         13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
3937
3938  4A12  E1               10.0  GENOUT: POP     H               ;RESTORE REGS
3939  4A13  C1               10.0          POP     B
3940  4A14  F1               10.0          POP     PSW
3941  4A15  C9               10.0          RET
3942
3943  4A16                                S
 (1)                                        ; *****************************************************************************
3944                                        ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
3945                                        ;       'CLK' BIT TO A '1' THEN A '0'.
3946  4A16                                S
 (1)                                        ; *****************************************************************************
3947
```

```
3948  4A16  3A  D5  48      13.0  CLKSYS: LDA   CCTLWD          ;GET SOFTWARE CLOCK CONTROL IMAGE
3949  4A19  F6  40             7.0          ORI   SSCLK          ;ADD IN 'CLK' BIT
3950  4A1B  D3  F0            10.0          OUT   CLKCTL         ;LOAD CLOCK CONTROL
3951  4A1D  E6  3F             7.0          ANI   @77            ;STRIP OFF CLOCK BIT
3952  4A1F  D3  F0            10.0          OUT   CLKCTL         ;LOAD CLOCK CONTROL WORD
3953  4A21  C9                 10.0          RET                  ;EXIT - CLOCK CYCLE COMPLETE
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*********************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*********************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*********************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************************
4A22  F5                        12.0    CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4A23  C5                        12.0            PUSH    B
4A24  06  00                     7.0            MVI     B,0             ;START TO CLEAR AT PORT #0
4A26  DB  E0                    10.0    CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4A28  E6  80                     7.0            ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4A2A  B0                         4.0            ORA     B               ;ADD IN THE SELECTED PORT #
4A2B  D3  E0                    10.0            OUT     MBSEL           ;RESET TO THIS PORT
4A2D  3E  80                     7.0            MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4A2F  D3  40                    10.0            OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4A31  AF                         4.0            XRA     A               ;CLEAR TU COMMAND A
4A32  D3  40                    10.0            OUT     TCMD            ;
4A34  3E  81                     7.0            MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4A36  D3  40                    10.0            OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4A38  3E  00                     7.0            MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
4A3A  D3  40                    10.0            OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
4A3C  AF                         4.0            XRA     A
4A3D  D3  44                    10.0            OUT     TAMT            ;CLEAR AMTIE WORD
4A3F  D3  48                    10.0            OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
4A41  D3  4C                    10.0            OUT     PENAB           ;CLEAR PORT ENABLE WORD
4A43  04                         4.0            INR     B               ;POINT TO THE NEXT PORT TO CLEAR
4A44  78                         4.0            MOV     A,B
4A45  FE  04                     7.0            CPI     4               ;DONE?
4A47  C2  26  4A                10.0            JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
4A4A  C1                        10.0            POP     B               ;RESET B & C
4A4B  F1                        10.0            POP     PSW             ;ALL DONE
4A4C  C9                        10.0            RET                     ;EXIT
      0000                              .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 0088          CBYTL  = 008A          CCTLWD  4805          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021           CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025           CH6TIE= 0026          CH7TIE= 0027
CKLOP  = 2815          CKSCLK  49B9           CLEAR   4A22          CLK     4834
CLKCNT  48DF           CLKCTL= 00F0           CLKER0  484D  G       CLKNXT  48B1
CLKON   48E0           CLKRS   4823           CLKSYS  4A16          CLK1    4832
CLOCK   4F26           CLRLP   4A26           CMC0H  = 0099          CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CRCWRD= 0018           CSAVE   4F9D          CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002
D      =%0002          DATACT= 00D0           DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRC0  = 0088
DDRCTL= 00DB           DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 0008          DONE1  = 0045          DONINT= 0010          DPNUM   48E1
DSAVE   4F9E           DSE    = 0006          D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003           ERCADR  48E6          ERFLG   4F93
ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812          ERLPE  = 280C
ERNUM   4F90           ERRCNT= 00D6           ESAVE   4F9F          EXDLPC  48D4
EXPADR  48DA           EXP00   4969           EXP01   496E          EXP02   4973
EXP03   4978           EXP04   497D           EXP05   4982          EXP06   4987
EXP07   498C           EXP10   4991           EXP11   4996          EXP12   499B
EXP13   49A0           EXP14   49A5           EXP15   49AA          EXP16   49AF
EXP17   49B4           E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FEXIT   4856          FIFORD= 006A
FINDIT  4752           FORMAT  4F25           FOUND   4774          FPORT   4794
FWDTST= 0061           GCRID  = 0089          GCRSET= 0002          GENDAT  49FD
GENOUT  4A12           GOODTM= 0092           H      =%0004          HLSAVE  4FA0
IE     = 0008          INTSTA= 00E0           ITERA   4F9A          I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020          I7.5   = 0040
KCALL  = 005F          KCLR   = 007B          KDEP   = 003F          KENAB  = 0078
KEXAM   003E           KEYBRD= 00C8           KEY1   = 0078          KEY10  = 006D
```

```
KEY11 = 006E    KEY12 = 006F    KEY13 = 005C    KEY14 = 005D
KEY15 = 005E    KEY16 = 005F    KEY17 = 003C    KEY18 = 003D
KEY19 = 003E    KEY2  = 0079    KEY20 = 003F    KEY3  = 007A
KEY4  = 007B    KEY5  = 0074    KEY6  = 0075    KEY7  = 0076
KEY8  = 0077    KEY9  = 006C    KINTA = 006F    KLDAD = 003D
KN0   = 003C    KN1   = 005C    KN2   = 005D    KN3   = 005E
KN4   = 006C    KN5   = 006D    KN6   = 006E    KN7   = 0074
KN8   = 0075    KN9   = 0076    KU2   = 0079    KU3   = 007A
KU8   = 0077    L     =%0005    LBLANK= 000F    LCE   = 000B
LCH   = 000C    LCL   = 000D    LCP   = 000E    LC0   = 0000
LC1   = 0001    LC2   = 0002    LC3   = 0003    LC4   = 0004
LC5   = 0005    LC6   = 0006    LC7   = 0007    LC8   = 0008
LC9   = 0009    LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC
LDLEDD= 00CD    LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800
LKKBD = 004C    LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055
LKLWPG= 0052    LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046
LPCADR  48E3    LPFLG   4F94    LPNUM   4F92    M     =%0006
MBSEL = 00E0    MB.A  = 0008    MB.B  = 0004    MEMTOP= 4FFF
MINUS = 000A    MM    = 8000    MSE   = 0008    MSGN  = 0013
MTACLR= C000    MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001
MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002
MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001
MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010    MT.Z  = 0008
M.ATA = 0080    M.CAPE= 0020    M.CONT= 0080    M.DEM = 0020
M.EBL = 0010    M.EXC = 0008    M.FAIL= 0008    M.ILR = 0010
M.INIT= 0010    M.OCC = 0020    M.ONLI= 0001    M.PE  = 0040
M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008    M.RUN = 0004
M.SCLK= 0001    M.TRA = 0040    M.UNIT= 0007    M.WCLK= 0040
M.WCLN= 0080    M.WREN= 0080    M5.5  = 0001    M6.5  = 0002
M7.5  = 0004    NOCLKC  48C9  G NOTCAP= 0088    NOUNIT  476B  G
NXTBYT  48A4    NXTB1   48B3    OKAY  = 00FF    OPRRAM= 4300
OPSTRT= 0058    OPVER = 0040    PADCNT= 00D5    PADCRC= 0080
PDIAG = 0048    PEID  = 008A    PENAB = 004C    PESET = 0001
PL    = 00B1    PRDD  = 004C    PRENF = 009C    PS    = 00B2
PSTAT = 0048    PSW   =%0009    P.AMTP= 0001    P.BCTC= 0040
P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040    P.LWR = 0020
P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002    P.RP0E= 0020
P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080
P.STAT= 0002    P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010
P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001    P.WPEN= 0010
P.WP0E= 0008    P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004
P.5VOK= 0002    QUE   = 281B    QUEM  = 281E    RAMT  = 0010
RARA  = 0006    RARAI = 0004    RCHBD0= 0048    RCHBD1= 0047
RCHOK = 0046    RCHTST= 000C    RCLRT = 000D    RCMD  = 000B
RCMLP = 0003    RCONT = 0080    RDATA = 0017    RDCLK = 0010
RDON  = 0011    READG = 0007    REND  = 0014    REQST = 2806
RESCHR= 00D1    REVTST= 0064    REWIND= 0004    RFIFOL= 0008
RGCLK = 0002    RGCRI = 0003    RIBG  = 0001    RILL  = 0012
RINST = 000C    RMCTST= 0008    RMK2  = 0013    RNOP  = 0000
RPATH = 0001    RPBAD = 0044    RPCHI = 0001    RPCLK = 0003
RPCTL = 0009    RPEI  = 0002    RPFAIL= 0000    RPF1  = 009D
RPF2  = 009E    RPOK  = 0043    RPOSTN= 0016    RPSTA = 0015
RRCMT = 000A    RSTAT = 0002    RTIEB = 000A    RTIER = 0030
RTM   = 0005    RUNKI = 0009    RUPTST= 005E    RWDUNL= 0005
```

```
R.AMT  = 0001    R.BOP  = 0008    R.DATA= 0040    R.DON  = 0002
R.DRDY= 0010    R.END  = 0010    R.ILL  = 0004    R.JVOK= 0004
R.MK2  = 0008    R.PLOD= 0008    R.PLO0= 0010    R.PLO1= 0020
R.POST= 0020    R.STNM= 0002    R.STOP= 0004    R.STPC= 0001
R.TBJN= 0080    R.TSTD= 0040    R.VOK  = 0080    R00H   = 0081
R00L   = 0080    R01H   = 0083    R01L   = 0082    R02H   = 0085
R02L   = 0084    R03H   = 0087    R03L   = 0086    R04H   = 0089
R04L   = 0088    R05H   = 0088    R05L   = 008A    R06H   = 008D
R06L   = 008C    R07H   = 008F    R07L   = 008E    R10H   = 0091
R10L   = 0090    R11H   = 0093    R11L   = 0092    R12H   = 0095
R12L   = 0094    R13H   = 0097    R13L   = 0096    R14H   = 0099
R14L   = 0098    R15H   = 009B    R15L   = 009A    R16H   = 009D
R16L   = 009C    R17H   = 009F    R17L   = 009E    R7.5   = 0010
SCKB1   49D3    SCLKBI  49C8    SDDRA   48DD    SDDRAT  48DC
SDDRB   48DE    SELCLR= 0000    SETATA= 00A1    SETDDR  49E1
SID    = 0080    SOD    = 0080    SOE    = 0040    SP     =%0008
SSCLK  = 0040    SSTEP  = 0005    STACK  = 4FFF    STATRM= 4F20
STPCT   4F20    STRSP  = 5000    SUPADR  48D8    SUP00   4929
SUP01   492D    SUP02   4931    SUP03   4935    SUP04   4939
SUP05   493D    SUP06   4941    SUP07   4945    SUP10   4949
SUP11   494D    SUP12   4951    SUP13   4955    SUP14   4959
SUP15   495D    SUP16   4961    SUP17   4965    SWMCST  48E2
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT   = 0044    TASEL  = 0080    TCMD   = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TEST10  44E2    TEST11  4527
TEST12  456C    TEST13  45B1    TEST14  45F6    TEST15  463B
TEST16  4680    TEST17  46C5    TEST2   4344    TEST20  470A
TEST3   4389    TEST4   43CE    TEST5   4413    TEST6   4458
TEST7   449D    TMF    = 0099    TMRDY  = 0040    TRANSC  48D3
TRKENA= 00D2    TSET   = 2803    TSTEND= 2818    TSTRST  47E8
TSTS   = 0040    TUSEL0= 00D1    TUSEL1= 00D2    TU78   = 0010
T.ATH0= 0001    T.ATH1= 0002    T.BOT  = 0004    T.EOT  = 0002
T.FPT  = 0001    T.NTHR= 0004    T.ONL  = 0020    T.PES  = 0008
T.PSBJ= 0020    T.PSOJ= 0008    T.PS1J= 0010    T.RDY  = 0080
T.RDYO= 0040    T.RWD  = 0010    T.SCLK= 0002    T1CONT  432A
T1DE0   4324  G  T1LOOP  4319    T1RPT   4305    T10CON  450D
T10DE0  4507  G  T10L00  44FC    T10RPT  44E7    T11CON  4552
T11DE0  454C  G  T11L00  4541    T11RPT  452C    T12CON  4597
T12DE0  4591  G  T12L00  4586    T12RPT  4571    T13CON  45DC
T13DE0  45D6  G  T13L00  45CB    T13RPT  45B6    T14CON  4621
T14DE0  461B  G  T14L00  4610    T14RPT  45FB    T15CON  4666
T15DE0  4660  G  T15L00  4655    T15RPT  4640    T16CON  46AB
T16DE0  46A5  G  T16L00  469A    T16RPT  4685    T17CON  46F0
T17DE0  46EA  G  T17L00  46DF    T17RPT  46CA    T2CONT  436F
T2DE0   4369  G  T2LOOP  435E    T2RPT   4349    T20CON  4735
T20DE0  472F  G  T20L00  4724    T20RPT  470F    T3CONT  43B4
T3DE0   43AE  G  T3LOOP  43A3    T3RPT   438E    T4CONT  43F9
T4DE0   43F3  G  T4LOOP  43E8    T4RPT   43D3    T5CONT  443E
T5DE0   4438  G  T5LOOP  442D    T5RPT   4418    T6CONT  4483
T6DE0   447D  G  T6LOOP  4472    T6RPT   445D    T7CONT  44C8
T7DE0   44C2  G  T7LOOP  44B7    T7RPT   44A2    UIBG   = 00A1
```

```
UNITMP  48D7        VALFC   4F98        VALTB   4F95        VELTST= 005B
WDR.P = 0010        WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0
WRTCLK= 0000        WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008
W.DIAG= 0002        W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080
W.ERR = 0020        W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004
W.ONES= 0020        W.RESI= 0002        W.REV = 0004        W.ROME= 0010
W.RST = 0001        W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020
X     =%000A        XDLOOP  4889        XDLP1   4896        XFRNUM  48D6
XFRSUB  482F        XFRTBL  48E9        XMCDAT  485B        X.DONN= 0080
X.ENAB= 0040        X.PEPE= 0002        X.ROME= 0001        X.WCLK= 0001
Y     =%000B        .     = 4A4D


 ERRORS DETECTED:  0

 *XMC5.A78/PTP,XMC5=NLIST,PARAM,MACRO,LIST,XMC5
 RUN-TIME: 6 11 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:11 SEQ 0581
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  XMC6 - TRANSLATOR MICRO CONTROLLER PART #6
                                .SBTTL  TEST 1 - XMC - RESYNC BURST TEST
                                :ID     XMC6-TRANSLATOR MICRO CONTROLLER PART #6
4300                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - RESYNC BURST TEST
4300                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A GCR RECORD OF 1113(10) BYTES
                                ;AND THEN SINGLE STEPS THE WRITE PATH (WMC, XMC) WHILE MONITORING THE
                                ;XMC ''XL WCLK'' BIT.  THE OUTPUT OF THE TRANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, 158 DATA GROUPS, RESYNC BURST, DATA GROUP, END MARK, RESIDUAL GROUP,
                                ;CRC GROUP, AND POSTAMBLE.  THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT,
                                ;*DATA STATES AND FLUX REVERSAL RATE.
4300                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      INITIALIZE THE BYTE COUNTER, PAD COUNTER, AND ERROR COUNTER
                                ;*      CLEAR THE ERROR COUNTER
                                ;*      REQUEST A TU PORT FROM THE HOST
                                ;*      CALL SUBROUTINE CLEAR
                                ;*      IF NO UNITS SELECTED
                                ;*      :     THEN-ERROR-ABORT TEST
                                ;*      :     ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      SET LOOP COMMAND/STATUS ON SELECTED PORT
                                ;*      CLEAR THE COMMAND BYTE FOR SELECTED PORT
                                ;*      SET WRITE PATH ENABLE
                                ;*      INITIALIZE THE EXPECTED RESULTS TABLE POINTER
                                ;*      SET WMC/XMC RESTART
                                ;*      SELECT THE SINGLE STEP CLOCK
                                ;*      CLEAR WMC/XMC RESTART
                                ;*      LOAD THE BYTE COUNT WITH 1113 (BITS 15:0)
                                ;*      SET DDR CONTROL TO ''IN''
                                ;*      SELECT THE IMAGE MODE OF DATA FORMATTING
                                ;*      LOAD THE RESIDUAL CHARACTER WITH 6
                                ;*      LOAD THE PAD COUNT WITH 6 (BITS 7:0)
                                ;*      CLEAR THE PAD COUNT (BITS 15:8)
                                ;*      ENABLE ALL TRACKS LOAD TRACK ENABLE WITH ONES
                                ;*      SET THE CLOCK COUNT TO 180(10)
                                ;*      BGNDO
                                ;*      :     CALL SUBROUTINE CLKSYS
                                ;*      :     DECREMENT THE CLOCK COUNT
                                ;*      :     DO UNTIL CLOCK COUNT=0
                                ;*      ENDDO
                                ;*      SET WRITE MICRO TO WRITE MICRO ENABLE, TRANSLATOR ENABLE, GCR, AND WRITE
                                ;*      SET THE CLOCK COUNT TO 256(10)
                                ;*      BGNDO
```

```
1374                    ;*      :     CALL SUBROUTINE CLKSYS
1375                    ;*      :     DECREMENT THE CLOCK COUNT
1376                    ;*      :     DO UNTIL CLOCK COUNT=0 OR XL WRITE CLOCK=1
1377                    ;*      ENDDO
1378                    ;*      IF CLOCK COUNT=0
1379                    ;*      :     THEN-FATAL ERROR-ABORT TEST
1380                    ;*      :     ELSE-CONTINUE
1381                    ;*      ENDIF
1382                    ;*      CLEAR THE GROUP COUNTER
1383                    ;*      BGNDO
1384                    ;*      :     CALL SUBROUTINE CLKSYS
1385                    ;*      :     CALL SUBROUTINE CLKSYS
1386                    ;*      :     INPUT THE TRANSLATOR OUTPUT DATA
1387                    ;*      :     IF THE ACTUAL TRANSLATOR OUTPUT=EXPECTED TRANSLATOR OUTPUT
1388                    ;*      :     :     THEN-CONTINUE
1389                    ;*      :     :     ELSE-ERROR
1390                    ;*      :     ENDIF
1391                    ;*      :     INPUT THE TRANSLATOR OUTPUT DATA PARITY
1392                    ;*      :     IF DATA OUTPUT DURING DATA GROUP
1393                    ;*      :     :     THEN-EXPECTED PARITY=ODD PARITY FOR OUTPUT DATA
1394                    ;*      :     :     ELSE-EXPECTED PARITY=1
1395                    ;*      :     ENDIF
1396                    ;*      :     IF INPUT PARITY=EXPECTED PARITY
1397                    ;*      :     :     THEN-CONTINUE
1398                    ;*      :     :     ELSE-ERROR
1399                    ;*      :     ENDIF
1400                    ;*      :     SET THE LOOP COUNT TO 14(8)
1401                    ;*      :     BGNDO
1402                    ;*      :     :     CALL SUBROUTINE CLKSYS
1403                    ;*      :     :     DECREMENT THE LOOP COUNT
1404                    ;*      :     :     DO UNTIL THE LOOP COUNT = 0
1405                    ;*      :     ENDDO
1406                    ;*      :     IF XL WRITE CLOCK = 1
1407                    ;*      :     :     THEN - CONTINUE
1408                    ;*      :     :     ELSE - ERROR - IMPROPER TRANSITION RATE FOR GCR
1409                    ;*      :     ENDIF
1410                    ;*      :     INCREMENT THE EXPECTED RESULTS TABLE POINTER
1411                    ;*      :     INCREMENT THE GROUP COUNTER
1412                    ;*      :     IF GROUP COUNTER < 158
1413                    ;*      :     :  THEN-DECREMENT THE EXPECTED RESULTS TABLE POINTER BY ONE GROUP.
1414                    ;*      :     :  ELSE-CONTINUE
1415                    ;*      :     ENDIF
1416                    ;*      :     DO UNTIL THE EXPECTED RESULTS POINTER=END OF TABLE
1417                    ;*      ENDDO
1418                    ;*      SET THE CLOCK COUNT TO 256(10)
1419                    ;*      BGNDO
1420                    ;*      :     CALL SUBROUTINE CLKSYS
1421                    ;*      :     DECREMENT THE CLOCK COUNT
1422                    ;*      :     DO UNTIL CLOCK COUNT=0 OR TRANSLATOR OR WRITE MICRO DONE=0
1423                    ;*      ENDDO
1424                    ;*      IF CLOCK COUNT=0
1425                    ;*      :     THEN-ERROR-TIMEOUT WAITING FOR WRITE MICRO DONE
1426                    ;*      :     ELSE-CONTINUE
1427                    ;*      ENDIF
```

```
1428                                    ;*    IF TRANSLATOR DONE
1429                                    ;*    :   THEN-ERROR-TRANSLATOR DONE BEFORE WRITE MICRO
1430                                    ;*    :   ELSE-SET THE CLOCK COUNT TO 256(10)
1431                                    ;*    :   BGNDO
1432                                    ;*    :   :   CALL SUBROUTINE CLKSYS
1433                                    ;*    :   :   DECREMENT THE CLOCK COUNT
1434                                    ;*    :   :   DO UNTIL TRANSLATOR DONE OR CLOCK COUNT=0
1435                                    ;*    :   ENDDO
1436                                    ;*    :   IF CLOCK COUNT=0
1437                                    ;*    :   :   THEN-ERROR-TIMEOUT WAITING FOR TRANSLATOR DONE
1438                                    ;*    :   :   ELSE-CONTINUE
1439                                    ;*    :   ENDIF
1440                                    ;*    ENDIF
1441                                    ;*ENDTST
1442  4300                              S
 (1)                                    ; ***************************************************************************************
1443
```

```
1445  4300                                  TEST1:  TESTX   @1
 (1)  4300  3E  01              7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                CALL    TSET            ;SETUP THE TEST
1446                                  ;%GCR - PREAMBLE/POSTAMBLE TEST - 1 DATA GROUP
1447                                  ;&M8958, M8959, M8955'S
1448
1449  4305  3E  01              7.0  XLLP1:  MVI     A,W.RST         ;CLEAR THE XMC ENABLE BIT
1450  4307  D3  D3             10.0          OUT     WMCCTL          ;DISABLE THE XMC
1451  4309  3E  00              7.0          MVI     A,0
1452  430B  D3  F0             10.0          OUT     CLKCTL          ;SET CLOCK CONTROL TO NORMAL OPERATION
1453  430D  AF                  4.0          XRA     A
1454  430E  D3  D3             10.0          OUT     WMCCTL          ;FINISH THE RESTART OF WMC/XMC
1455                                                                  ;XMC SHOULD START SELF TEST
1456  4310  3E  FF              7.0          MVI     A,@377          ;LOAD A COUNT VALUE
1457  4312  00                  4.0  1$:     NOP                     ;NOP
1458  4313  3D                  4.0          DCR     A               ;DECREMENT COUNT
1459  4314  C2  12  43          10.0          JNZ     1$              ;CONTINUE UNTIL COUNT = 0
1460                                                                  ;
1461
1462  4317  AF                  4.0          XRA     A               ;CLEAR UNIT MAP STORAGE
1463  4318  32  25  47          13.0          STA     UNITMP          ;1ST TIME HERE
1464  431B  32  29  47          13.0          STA     EXDLPC          ;CLEAR THE ERROR COUNTER
1465
1466  431E  3E  30              7.0  T1LOOP: MVI     A,@60           ;INIT THE BYTE COUNTER
1467  4320  D3  D7             10.0          OUT     CNTCTL          ;TO NORMAL MODE
1468  4322  3E  70              7.0          MVI     A,@160          ;INIT THE PAD COUNTER
1469  4324  D3  D7             10.0          OUT     CNTCTL          ;TO NORMAL MODE
1470  4326  3E  B0              7.0          MVI     A,@260          ;INIT THE ERROR COUNTER
1471  4328  D3  D7             10.0          OUT     CNTCTL          ;TO NORMAL MODE
1472  432A                                  CLRECT
 (1)  432A  AF                  4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  432B  D3  D6             10.0                  OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  432D  D3  D6             10.0                  OUT     ERRCNT          ;CLEAR BITS 15-8
```

```
                                        ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
                                        ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.

 432F  3A 25 47      13.0       LDA     UNITMP          ;GET THE UNIT MAP STORAGE
 4332  A7             4.0       ANA     A
 4333  C2 4C 43      10.0       JNZ     T1LPA           ;JUMP OVER IF ALREADY HAVE IT
 4336                           REQ     @7,0,0,0
(1) 4336  CD 06 28    18.0                 CALL    REQST
(1) 4339  00                               .BYTE   0               ;DATA PATTERN NUMBER
(1) 433A  00 00                            .WORD   0               ;SYSTEM "0" COUNT
(1) 433C  00 00                            .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(1) 433E  00                               .BYTE                   ;DATA COMPARE FLAG IF =1
(1) 433F  07                               .BYTE   @7              ;REQUEST CODE
 4340                           RIN     R12L
(1) 4340  DB 94       10.0                 IN      R12L                    ;READ R12L INTO AC
(1) 4342  7F           4.0                 MOV     A,A                     ;RETRY LINK
 4343  32 25 47      13.0       STA     UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
 4346  AF             4.0       XRA     A
 4347  CD C6 45      18.0       CALL    CLEAR
 434A  06 00            7.0    MVI     B,0
 434C  3A 25 47      13.0  T1LPA: LDA     UNITMP          ;GET SELECTED UNIT MAP
 434F  E6 01            7.0    ANI     $01             ;UNIT 0 PRESENT?
 4351  C2 7D 43      10.0       JNZ     FOUND           ;YES-USE IT
 4354  04             4.0       INR     B
 4355  3A 25 47      13.0       LDA     UNITMP          ;
 4358  E6 02            7.0    ANI     $02             ;UNIT 1 PRESENT?
 435A  C2 7D 43      10.0       JNZ     FOUND           ;YES-USE IT
 435D  04             4.0       INR     B
 435E  3A 25 47      13.0       LDA     UNITMP          ;
 4361  E6 04            7.0    ANI     $04             ;UNIT 2 PRESENT?
 4363  C2 7D 43      10.0       JNZ     FOUND           ;YES-USE IT
 4366  04             4.0       INR     B
 4367  3A 25 47      13.0       LDA     UNITMP
 436A  E6 08            7.0    ANI     $08             ;UNIT 3 PRESENT
 436C  C2 7D 43      10.0       JNZ     FOUND           ;YES-USE IT
                                                        ;ELSE - OPERATOR ERROR
 436F                           ERR     EXIT,DUMMY
(1)                             ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 436F  CD 09 28    18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)       0001                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 4372  01                               .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 4373  00                               .BYTE
(1) 4374  CD 15 28    18.0            DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1) 4377  DA B7 45    10.0                 JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
                        ;>OPERATOR ERROR NO TM78 UNITS SELECTED
                        ;>FATAL ERROR - TEST ABORTED
                        ;*XMC6 MICRO ERROR 01
                        ;*OPERATOR ERROR NO TM78 UNITS SELECTED
                        ;*FATAL ERROR - TEST ABORTED
 437A  C3 B7 45      10.0       JMP     EXIT
```

```
1510                                        ;SET UP FOR THE DATA TRANSFER
1511  437D  DB  E0          10.0  FOUND:  IN      INTSTA
1512  437F  E6  80           7.0          ANI     BIT7
1513  4381  B0               4.0          ORA     B
1514  4382  D3  E0          10.0          OUT     MBSEL
1515  4384  3E  40           7.0          MVI     A,P.LCS         ;KILL THE MTA BOARD
1516  4386  D3  48          10.0          OUT     PDIAG
1517  4388  AF               4.0          XRA     A
1518  4389  D3  40          10.0          OUT     TCMD
1519  438B  3E  10           7.0          MVI     A,P.WPEN        ;SET WRITE PATH ENABLE
1520  438D  D3  4C          10.0          OUT     PENAB
1521  438F  21  F1  45      10.0          LXI     H,GXPTBL        ;LOAD ADDRESS OF EXPECTED
1522  4392  22  2B  47      16.0          SHLD    POINTER         ;RESULTS TABLE INTO POINTER
1523  4395  3E  01           7.0          MVI     A,W.RST         ;LOAD THE WMC/XMC RESTART CODE
1524  4397  D3  D3          10.0          OUT     WMCCTL          ;ISSUE THE COMMAND
1525  4399  3E  05           7.0          MVI     A,SSTEP         ;SELECT SINGLE STEP WRITE CLOCK
1526  439B  D3  F0          10.0          OUT     CLKCTL          ;INITIALIZE THE CLOCK
1527  439D  32  23  47      13.0          STA     CCTLWD          ;SAVE SOFTWARE IMAGE
1528  43A0  AF               4.0          XRA     A               ;CLEAR THE ACC.
1529  43A1  D3  D3          10.0          OUT     WMCCTL          ;CLEAR THE WMC/XMC RESTART BIT
1530  43A3  21  59  04      10.0          LXI     H,1113          ;GET THE BYTE COUNT - 1 DATA GROUP
1531  43A6  23               6.0          INX     H               ;CORRECT FOR INTEL 8253 CHIP
1532  43A7  7D               4.0          MOV     A,L
1533  43A8  D3  D4          10.0          OUT     BYTCNT          ;LOAD BITS 7-0
1534  43AA  7C               4.0          MOV     A,H
1535  43AB  D3  D4          10.0          OUT     BYTCNT          ;CLEAR THE BYTE COUNTER BITS 15-8
1536  43AD  AF               4.0          XRA     A
1537  43AE  32  2D  47      13.0          STA     TRANSCNT        ;INITIALIZE THE TRANSITION COUNT
1538  43B1  32  2E  47      13.0          STA     TRANSCNT+1
1539  43B4  32  27  47      13.0          STA     DPCNT           ;CLEAR THE PARITY TEST COUNTER
1540  43B7  32  28  47      13.0          STA     DPCNT+1
1541  43BA  32  24  47      13.0          STA     GRPCNT
1542  43BD  3E  88           7.0          MVI     A,@210
1543  43BF  D3  DB          10.0          OUT     DDRCTL          ;SET THE DDR CONTROL TO 'IN'
1544  43C1  3E  50           7.0          MVI     A,$50           ;LOAD THE IMAGE MODE COMMAND
1545  43C3  D3  D0          10.0          OUT     DATACTL         ;SET THE DATA CONTROL MODE
1546  43C5  3E  18           7.0          MVI     A,24            ;LOAD THE RESIDUAL CHARACTER VALUE
1547  43C7  D3  D1          10.0          OUT     RESCHR          ;SET THE RESIDUAL CHARACTER
1548  43C9  3E  06           7.0          MVI     A,6             ;LOAD PAD COUNT VALUE
1549  43CB  3C               4.0          INR     A               ;CORRECT FOR INTEL 8253 CHIP
1550  43CC  D3  D5          10.0          OUT     PADCNT          ;WRITE TO WMC BITS 7-0
1551  43CE  AF               4.0          XRA     A               ;LOAD HI VALUE
1552  43CF  D3  D5          10.0          OUT     PADCNT          ;WRITE TO WMC
1553  43D1  D3  D2          10.0          OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS 1ST
1554  43D3  D3  D2          10.0          OUT     TRKENA
1555  43D5  3E  FF           7.0          MVI     A,$FF           ;LOAD ACC WITH ALL ONES
1556  43D7  D3  D2          10.0          OUT     TRKENA          ;ENABLE DATA TRACKS
1557  43D9  D3  D2          10.0          OUT     TRKENA          ;ENABLE PARITY TRACK
1558  43DB  06  B4           7.0          MVI     B,180           ;LOAD THE CLOCK COUNT
```

```
1560  43DD  CD  BA  45      18.0  CLKRS:  CALL   CLKSYS            ;CLOCK IT
1561  43E0  05                4.0          DCR    B                ;
1562  43E1  C2  DD  43       10.0          JNZ    CLKRS            ;CONTINUE WITH RESTART CLOCKING
1563
1564  43E4  3E  D8             7.0          MVI    A,W.ENAB!X.ENAB!W.GCR!W.WRITE
1565  43E6  D3  D3            10.0          OUT    WMCCTL           ;START THE OPERATION
1566  43E8  06  FF             7.0  CLK1:   MVI    B,$FF            ;SET TIMEOUT COUNT TO 256
1567                                                                ;
1568  43EA  CD  BA  45        18.0  CLK:    CALL   CLKSYS           ;SINGLE STEP THE WRITE PATH
1569                                                                ;
1570  43ED  DB  D0            10.0          IN     WMCSTA           ;GET THE WMC STATUS
1571  43EF  E6  01             7.0          ANI    X.WCLK           ;X.WCLK SET?
1572  43F1  C2  06  44        10.0          JNZ    BGNPRE           ;YES-BEGINNING OF PREAMBLE
1573  43F4  05                 4.0          DCR    B                ;NO-DECREMENT CLOCK
1574  43F5  C2  EA  43        10.0          JNZ    CLK              ;CONTINUE UNTIL ZERO
1575  43F8                                  ERR    TSTEND,ABOCN0
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43F8  CD  09  28        18.0                 CALL   ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0002                              MSGN   =      MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43FB  02                               .BYTE  MSGN             ;MESSAGE NUMBER ID
 (1)  43FC  00                               .BYTE
 (1)  43FD  CD  15  28        18.0  ABOCN0::       CALL   CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4400  DA  18  28        10.0                 JC     TSTEND           ;LOOP ADDRESS IF LOOP SPECIFIED
1576                                  ;>WRITE PATH TIMEOUT WHILE WAITING FOR 'XL CLK'' TO SET
1577                                  ;>FATAL ERROR - TEST ABORTED
1578                                  ;*XMC6 MICRO ERROR 02
1579                                  ;*WRITE PATH TIMEOUT WHILE WAITING FOR 'XL CLK'' TO SET
1580                                  ;*FATAL ERROR - TEST ABORTED
1581
1582  4403  C3  18  28        10.0          JMP    TSTEND
```

```
1584  4406  CD  BA  45   18.0  BGNPRE: CALL  CLKSYS          ;CLOCK THE SYSTEM
1585  4409  CD  BA  45   18.0          CALL  CLKSYS          ;CLOCK AGAIN
1586  440C  DB  40        10.0          IN    TSTS            ;GET THE DATA READ
1587  440E  47            4.0          MOV   B,A             ;SAVE IT IN B=ADATA
1588  440F                              ROUT  ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  440F  D3  94        10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  4411  7F            4.0                 MOV   A,A               ;RETRY LINK
1589                                                        ;
1590  4412  2A  2D  47   16.0          LHLD  TRANSCNT        ;GET THE TRANSITION COUNT
1591  4415  23            6.0          INX   H               ;UPDATE
1592  4416  22  2D  47   16.0          SHLD  TRANSCNT        ;RESTORE THE TRANSITION COUNT
1593  4419  7D            4.0          MOV   A,L             ;GET TRANSCOUNT LOW
1594  441A                              ROUT  R05L            ;WRITE TO CAS
 (1)  441A  D3  8A        10.0                OUT   R05L              ;WRITE AC INTO R05L
 (1)  441C  7F            4.0                 MOV   A,A               ;RETRY LINK
1595  441D  7C            4.0          MOV   A,H             ;GET TRANSCOUNT HIGH
1596  441E                              ROUT  R05H            ;WRITE TO CAS
 (1)  441E  D3  8B        10.0                OUT   R05H              ;WRITE AC INTO R05H
 (1)  4420  7F            4.0                 MOV   A,A               ;RETRY LINK
1597  4421  2A  2B  47   16.0          LHLD  POINTER         ;GET POINTER TO EXPECTED DATA
1598  4424  7E            7.0          MOV   A,M             ;GET EXPECTED DATA BYTE
1599  4425  32  26  47   13.0          STA   XDATA           ;SAVE FOR LATER
1600  4428  B8            4.0          CMP   B               ;COMPARE WITH ACTUAL
1601  4429  CA  41  44   10.0          JZ    PRECN1          ;CONTINUE IF EQUAL
1602                                                        ;ELSE-ERROR
1603  442C                              ROUT  EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  442C  D3  95        10.0                OUT   EDATA             ;WRITE AC INTO EDATA
 (1)  442E  7F            4.0                 MOV   A,A               ;RETRY LINK
1604  442F  3A  29  47   13.0  XDLPC1: LDA   EXDLPC          ;GET THE CURRENT COUNT AT ERROR
1605  4432  A7            4.0          ANA   A
1606  4433  C2  3C  44   10.0          JNZ   XDLPC2          ;STAY AT THIS COUNT IF NOT ZERO
1607  4436  2A  2D  47   16.0          LHLD  TRANSCNT        ;GET THE COUNT AT THE ERROR
1608  4439  22  29  47   16.0          SHLD  EXDLPC          ;SAVE IT
1609  443C                        XDLPC2: ERRB  XDLOOP,PRECN1,@4
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  443C  CD  12  28   18.0                CALL  ERLPB             ;PROCESS ERROR - DO 2.3
 (1)        0003                             MSGN  =     MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  443F  03                               .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  4440  04                               .BYTE @4                ;PRINT ROUTINE NUMBER
 (1)  4441  CD  15  28   18.0          PRECN1::      CALL  CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4444  DA  4A  44   10.0                JC    XDLOOP            ;LOOP ADDRESS IF LOOP SPECIFIED
1610                                ;>DATA STATE RECEIVED FROM THE XMC INCORRECT
1611                                ;*XMC6 MICRO ERROR 03
1612                                ;*DATA STATE RECEIVED FROM THE XMC INCORRECT
1613                                ;*ACTUAL = NNNN
1614                                ;*EXPECTED = NNNN
1615  4447  C3  6C  44   10.0          JMP   XDLPC           ;SKIP OVER ERROR LOOP CONTROL
1616
```

```
1618
1619                                        ;HERE IS THE ERROR LOOP CONTROL
1620
1621  444A  3A  94  4F    13.0  XDLOOP: LDA   LPFLG     ;GET THE LOOP ON ERROR FLAG
1622  444D  A7                   4.0        ANA   A
1623  444E  C2  5A  44          10.0        JNZ   XDLP1     ;JUMP IF FLAG IS SET
1624  4451  21  00  00          10.0        LXI   H,0       ;CLEAR H&L
1625  4454  22  29  47          16.0        SHLD  EXDLPC    ;CLEAR THE ERROR POINTER
1626  4457  C3  6C  44          10.0        JMP   XDLPC     ;AND CONTINUE TESTING
1627
1628  445A  2A  2D  47          16.0  XDLP1:  LHLD  TRANSCNT  ;GET THE CURRENT XLWRITE CLOCK COUNT
1629  445D  EB                   4.0        XCHG            ;PUT H&L IN D&E
1630  445E  2A  29  47          16.0        LHLD  EXDLPC    ;GET THE FAILING COUNT
1631  4461  7A                   4.0        MOV   A,D
1632  4462  2F                   4.0        CMA
1633  4463  57                   4.0        MOV   D,A
1634  4464  7B                   4.0        MOV   A,E
1635  4465  2F                   4.0        CMA
1636  4466  5F                   4.0        MOV   E,A
1637  4467  13                   6.0        INX   D
1638  4468  19                  10.0        DAD   D         ;CURRENT COUNT = FAILING COUNT?
1639  4469  DA  1E  43          10.0        JC    T1LOOP    ;YES - LOOP NOW
```

```
1641  446C  DB  48        10.0  XDLPC:  IN      PSTAT           ;GET THE PORT PARITY BIT
1642  446E  0F              4.0          RRC                     ;RIGHT JUSTIFY
1643  446F  0F              4.0          RRC                     ;
1644  4470  E6  01          7.0          ANI     $01             ;REMOVE OTHER BITS
1645  4472  57              4.0          MOV     D,A             ;STORE ACTUAL PARITY IN REG D
1646
1647  4473                                ROUT    ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  4473  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4475  7F              4.0                   MOV     A,A             ;RETRY LINK
1648  4476  2A  2B  47      16.0          LHLD    POiNTER
1649  4479  01  46  46      10.0          LXI     B,GXPDAT        ;LOAD BEGINNING ADDRESS OF EXPECTED DATA
1650  447C  78              4.0          MOV     A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1651  447D  2F              4.0          CMA                     ;PAIR B & C
1652  447E  47              4.0          MOV     B,A
1653  447F  79              4.0          MOV     A,C
1654  4480  2F              4.0          CMA
1655  4481  4F              4.0          MOV     C,A
1656  4482  03              6.0          INX     B
1657  4483  09             10.0          DAD     B               ;ADD POINTER AND BEGINNING OF DATA TABLE
1658  4484  D2  AB  44      10.0          JNC     PPREAM          ;CHECK PREAMBLE PARITY TRACK
1659
1660  4487  2A  2B  47      16.0          LHLD    POINTER         ;RESET H & L REGS
1661  448A  01  8C  46      10.0          LXI     B,GDATEN        ;GET THE ENDING ADDRESS OF EXPECTED DATA
1662  448D  78              4.0          MOV     A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1663  448E  2F              4.0          CMA                     ;PAIR B & C
1664  448F  47              4.0          MOV     B,A
1665  4490  79              4.0          MOV     A,C
1666  4491  2F              4.0          CMA
1667  4492  4F              4.0          MOV     C,A
1668  4493  03              6.0          INX     B
1669  4494  09             10.0          DAD     B               ;ADD POINTFR AND END OF DATA TABLE
1670  4495  DA  AB  44      10.0          JC      PPOST           ;CHECK POSTAMBLE PARITY TRACK
1671
1672
1673
1674  4498  01  DC  46      10.0  DPTST:  LXI     B,PXPTBL        ;POINT TO EXPECTED PARITY DATA TABLE
1675  449B  2A  27  47      16.0          LHLD    DPCNT           ;POINT TO THE TABLE OFFSET NUMBER
1676  449E  23              6.0          INX     H               ;UPDATE IT
1677  449F  22  27  47      16.0          SHLD    DPCNT           ;SAVE NEW COUNT
1678  44A2  09             10.0          DAD     B               ;MAKE REAL POINTER
1679  44A3  7E              7.0          MOV     A,M             ;GET EXPECTED PARITY STATE
1680  44A4  A7              4.0          ANA     A               ;SET CONDITION BITS
1681  44A5  FA  F8  44      10.0          JM      PTYC            ;JMP IF NO COMPARE ON THIS DATA
1682  44A8  C3  B0  44      10.0          JMP     PTY             ;TEST THE PARITY BIT
1683
1684                                ;HERE TO CHECK THE PREAMBLE/POSTAMBLE PARITY TRACK BIT
1685
1686  44AB                        PPOST:
1687  44AB  3A  26  47      13.0  PPREAM: LDA     XDATA           ;GET THE DATA EXPECTED
1688  44AE  E6  01           7.0          ANI     1               ;PARITY TRACK IS SAME
```

```
1690  44B0  BA              4.0  PTY:   CMP   D                ;SEE IF EXPECTED=ACTUAL
1691  44B1  CA  CD  44       10.0         JZ    PTSTOK           ;JMP IF OK
1692  44B4                           ROUT  EDATA            ;SAVE THE EXPECTED DATA
 (1)  44B4  D3  95            10.0         OUT   EDATA              ;WRITE AC INTO EDATA
 (1)  44B6  7F                 4.0         MOV   A,A                ;RETRY LINK
1693  44B7  7A                 4.0         MOV   A,D          ;ERROR - GET THE ACTUAL DATA
1694  44B8                           ROUT  ADATA            ;SAVE FOR PRINTOUT
 (1)  44B8  D3  94            10.0         OUT   ADATA              ;WRITE AC INTO ADATA
 (1)  44BA  7F                 4.0         MOV   A,A                ;RETRY LINK
1695
1696  44BB  3A  29  47         13.0  PTY1:  LDA   EXDLPC           ;GET THE COUNT AT THE ERROR
1697  44BE  A7                 4.0         ANA   A
1698  44BF  C2  C8  44         10.0         JNZ   PTY2             ;JUMP IF AT AN ERROR NOW
1699  44C2  2A  2D  47         16.0         LHLD  TRANSCNT         ;GET THE COUNT AT ERROR
1700  44C5  22  29  47         16.0         SHLD  EXDLPC           ;SAVE IT
1701  44C8                           PTY2:  ERRB  XPLOOP,PTSTOK,@4
 (1)                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44C8  CD  12  28         18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0004                      MSGN  =     MSGN+1                 ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CB  04                        .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  44CC  04                        .BYTE @4               ;PRINT ROUTINE NUMBER
 (1)  44CD  CD  15  28         18.0  PTSTOK::       CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D0  DA  D6  44         10.0                JC    XPLOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1702                       ;>PARITY STATE RECEIVED FROM THE XMC INCORRECT
1703                       ;*XMC6 MICRO ERROR 04
1704                       ;*PARITY STATE RECEIVED FROM THE XMC INCORRECT
1705                       ;*ACTUAL = NNNN
1706                       ;*EXPECTED = NNNN
1707  44D3  C3  F8  44         10.0         JMP   PTYC             ;JMP OVER CODE
1708
1709
1710
1711  44D6  3A  94  4F         13.0  XPLOOP: LDA   LPFLG          ;GET THE LOOP ON ERROR FLAG
1712  44D9  A7                 4.0         ANA   A
1713  44DA  C2  E6  44         10.0         JNZ   XPLP1            ;JUMP IF FLAG IS SET
1714  44DD  21  00  00         10.0         LXI   H,0              ;CLEAR H&L
1715  44E0  22  29  47         16.0         SHLD  EXDLPC           ;CLEAR THE ERROR POINTER
1716  44E3  C3  F8  44         10.0         JMP   PTYC             ;AND CONTINUE TESTING
1717
1718  44E6  2A  2D  47         16.0  XPLP1:  LHLD  TRANSCNT        ;GET THE CURRENT XLWRITE CLOCK COUNT
1719  44E9  EB                 4.0         XCHG                   ;PUT H&L IN D&E
1720  44EA  2A  29  47         16.0         LHLD  EXDLPC           ;GET THE FAILING COUNT
1721  44ED  7A                 4.0         MOV   A,D
1722  44EE  2F                 4.0         CMA
1723  44EF  57                 4.0         MOV   D,A
1724  44F0  7B                 4.0         MOV   A,E
1725  44F1  2F                 4.0         CMA
1726  44F2  5F                 4.0         MOV   E,A
1727  44F3  13                 6.0         INX   D
1728  44F4  19                10.0         DAD   D                ;CURRENT COUNT = FAILING COUNT?
1729  44F5  DA  1E  43         10.0         JC    T1LOOP           ;YES - LOOP NOW
```

```
1731  44F8  2A  2B  47    16.0  PTYC:  LHLD   POINTER          ;NO-UPDATE THE
1732  44FB  23                 6.0         INX    H              ;TABLE POINTER
1733  44FC  22  2B  47       16.0         SHLD   POINTER        ;ADDRESS
1734  44FF  2A  2B  47       16.0         LHLD   POINTER
1735  4502  01  50  46       10.0         LXI    B,RESYNC
1736  4505  78                 4.0         MOV    A,B
1737  4506  2F                 4.0         CMA
1738  4507  47                 4.0         MOV    B,A
1739  4508  79                 4.0         MOV    A,C
1740  4509  2F                 4.0         CMA
1741  450A  4F                 4.0         MOV    C,A
1742  450B  03                 6.0         INX    B
1743  450C  09                10.0         DAD    B              ;IS A GROUP DONE
1744  450D  D2  29  45        10.0         JNC    CONT           ;NO-CONTINUE
1745  4510  3A  24  47        13.0         LDA    GRPCNT         ;YES-GET GROUP COUNT
1746  4513  3C                 4.0         INR    A              ;GROUP COUNT
1747  4514  FE  9E              7.0         CPI    158            ;END OF GROUP 158?
1748  4516  CA  29  45        10.0         JZ     CONT           ;YES-CONTINUE
1749  4519  32  24  47        13.0         STA    GRPCNT
1750  451C  21  46  46        10.0         LXI    H,GXPDAT
1751  451F  22  2B  47        16.0         SHLD   POINTER
1752  4522  AF                 4.0         XRA    A
1753  4523  32  27  47        13.0         STA    DPCNT
1754  4526  32  28  47        13.0         STA    DPCNT+1
1755  4529  06  0E              7.0  CONT:  MVI    B,14           ;LOAD THE CLOCK COUNT
1756  452B  CD  BA  45        18.0  1$:    CALL   CLKSYS         ;CLOCK THE WRITE PATH
1757  452E  05                 4.0         DCR    B              ;DECREMENT THE COUNT
1758  452F  C2  2B  45        10.0         JNZ    1$             ;CONTINUE UNTIL 8 CLOCKS
1759  4532  DB  D0            10.0         IN     WMCSTA
1760  4534  E6  01              7.0         ANI    X.WCLK
1761  4536  C2  3E  45        10.0         JNZ    PRECN3
1762  4539                                 ERR    T1LOOP,PRECN3,@4
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4539  CD  09  28        18.0                CALL   ERLP           ;PROCESS ERROR - DO 2.3
 (1)        0005                          MSGN   =      MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453C  05                          .BYTE  MSGN           ;MESSAGE NUMBER ID
 (1)  453D  04                          .BYTE  @4
 (1)  453E  CD  15  28        18.0  PRECN3::      CALL   CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4541  DA  1E  43        10.0                JC     T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1763                                 ;>TRANSLATOR 'XL WCLK' NOT SET AFTER EIGHT CLOCK CYCLES
1764                                 ;>IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1765                                 ;*XMC6 MICRO ERROR 05
1766                                 ;*TRANSLATOR 'XL WCLK' NOT SET AFTER EIGHT CLOCK CYCLES
1767                                 ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
```

```
1769  4544  2A  2B  47   16.0          LHLD   POINTER          ;GET THE TABLE POINTER
1770  4547  01  DB  46   10.0          LXI    B,GTBLEN         ;GET END OF TABLE ADDRESS
1771  454A  78               4.0       MOV    A,B              ;FROM THE
1772  454B  2F               4.0       CMA                     ;TWO'S COMPLIMENT
1773  454C  47               4.0       MOV    B,A              ;OF THE TABLE END
1774  454D  79               4.0       MOV    A,C              ;ADDRESS
1775  454E  2F               4.0       CMA
1776  454F  4F               4.0       MOV    C,A
1777  4550  03               6.0       INX    B
1778  4551  09              10.0       DAD    B                ;END OF TABLE?
1779  4552  DA  58  45      10.0       JC     RECEND           ;YES
1780  4555  C3  06  44      10.0       JMP    BGNPRE
1781
1782  4558  06  FF           7.0  RECEND: MVI    B,$FF            ;LOAD THE TIMEOUT COUNT
1783  455A  CD  BA  45      18.0  1$:    CALL   CLKSYS           ;CLOCK THE SYSTEM
1784  455D  DB  D0           10.0       IN     WMCSTA           ;INPUT THE WMC STATUS
1785  455F  E6  40            7.0       ANI    W.DONN           ;WMC DONE?
1786  4561  CA  74  45      10.0       JZ     RECCN1           ;YES-EXIT
1787  4564  DB  D0           10.0       IN     WMCSTA           ;GET WMC STATUS
1788  4566  E6  80            7.0       ANI    X.DONN           ;XMC DONE
1789  4568  CA  98  45      10.0       JZ     SEQERR           ;YES-ERROR
1790  456B  05               4.0       DCR    B                ;NO-DECREMENT THE COUNT
1791  456C  C2  5A  45      10.0       JNZ    1$               ;CONTINUE IF NOT ZERO
1792  456F                             ERR    RECEND,RECCN1
  (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  456F  CD  09  28      18.0                CALL   ERLP             ;PROCESS ERROR - DO 2.3
  (1)        0006                    MSGN   =      MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4572  06                      .BYTE  MSGN             ;MESSAGE NUMBER ID
  (1)  4573  00                      .BYTE
  (1)  4574  CD  15  28      18.0  RECCN1::      CALL   CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4577  DA  58  45      10.0                JC     RECEND           ;LOOP ADDRESS IF LOOP SPECIFIED
1793                       ;>'WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1794                       ;*XMC6 MICRO ERROR 06
1795                       ;*'WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1796
1797  457A  06  FF           7.0       MVI    B,$FF            ;LOAD THE TIMEOUT COUNT
1798  457C  CD  BA  45      18.0  3$:    CALL   CLKSYS           ;CLOCK THE SYSTEM
1799  457F  DB  D0           10.0       IN     WMCSTA           ;INPUT THE WMC STATUS
1800  4581  E6  80            7.0       ANI    X.DONN           ;XMC-DONE?
1801  4583  CA  8F  45      10.0       JZ     RECCN3           ;YES-EXIT
1802  4586  05               4.0       DCR    B                ;NO-DECREMENT THE COUNT
1803  4587  C2  7C  45      10.0       JNZ    3$               ;CONTINUE IF NOT ZERO
1804  458A                             ERR    T1LOOP,RECCN3
  (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  458A  CD  09  28      18.0                CALL   ERLP             ;PROCESS ERROR - DO 2.3
  (1)        0007                    MSGN   =      MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  458D  07                      .BYTE  MSGN             ;MESSAGE NUMBER ID
  (1)  458E  00                      .BYTE
  (1)  458F  CD  15  28      18.0  RECCN3::      CALL   CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4592  DA  1E  43      10.0                JC     T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1805                       ;>'XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1806                       ;*XMC6 MICRO ERROR 07
1807                       ;*'XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1808  4595  C3  A3  45      10.0       JMP    XMEND
```

```
1810  4598                              SEQERR: ERR     T1LOOP,RECCN2
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4598  CD  09  28        18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  459B  08                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459C  00                                 .BYTE
 (1)  459D  CD  15  28        18.0    RECCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45A0  DA  1E  43        10.0                JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1811                                    ;>''XMC NOT DONE'' RESET BEFORE 'WMC NOT DONE'' RESET
1812                                    ;*XMC6 MICRO ERROR 10
1813                                    ;*''XMC NOT DONE'' RESET BEFORE 'WMC NOT DONE'' RESET
1814
1815  45A3                              XMEND:  ENDTST  XLLP1
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45A3                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45A3  CD  06  28        18.0                CALL    REQST
 (2)  45A6  00                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  45A7  00  00                             .WORD                   ;SYSTEM '*' COUNT
 (2)  45A9  00  00                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45AB  00                                 .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  45AC  07                                 .BYTE   7               ;REQUEST CODE
 (1)  45AD  3A  9A  4F        13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  45B0  3D                 4.0                DCR     A               ;DOWNCOUNT
 (1)  45B1  32  9A  4F        13.0                STA     ITERA           ;SAVE COUNT
 (1)  45B4  F2  05  43        10.0                JP      XLLP1           ;DO TEST UNTIL TILL = 0
1816
1817  45B7  C3  18  28        10.0  EXIT:   JMP     TSTEND
1818
```

```
                                        .SBTTL  SUBROUTINE CLKSYS

                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE WILL GENERATE ONE SINGLE STEP USING THE CONTENTS OF
                                        ;*CCTLWD AS THE CLOCK CONTROL BYTE.
                                        S
                                        ; ************************************************************************

45BA 3A 23 47          13.0  CLKSYS: LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
45BD F6 40              7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
45BF D3 F0             10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
45C1 E6 BF              7.0          ANI     @277            ;STRIP OFF CLOCK BIT
45C3 D3 F0             10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
45C5 C9                10.0          RET                     ;EXIT-CLOCK CYCLE
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS

                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************

                                12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                                12.0          PUSH    B
                                 7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
```

```
1857  45CA  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
1858  45CC  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
1859  45CE  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1860  45CF  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
1861
1862  45D1  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
1863  45D3  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
1864  45D5  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
1865  45D6  D3  40          10.0          OUT     TCMD            ;
1866  45D8  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
1867  45DA  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
1868  45DC  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
1869  45DE  D3  40          10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
1870  45E0  AF               4.0          XRA     A
1871  45E1  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
1872  45E3  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
1873  45E5  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
1874
1875  45E7  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
1876  45E8  78               4.0          MOV     A,B
1877  45E9  FE  04           7.0          CPI     4               ;DONE?
1878  45EB  C2  CA  45      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
1879
1880  45EE  C1              10.0          POP     B               ;RESET B & C
1881  45EF  F1              10.0          POP     PSW             ;ALL DONE
1882  45F0  C9              10.0          RET                     ;EXIT
1883
```

```
1885
1886  45F1  FF          GXPTBL: .BYTE   $FF      ;PREAMBLE TERM GROUP (10101)
1887                                              ;CHARACTER 1 = ALL ONES
1888  45F2  FF                  .BYTE   $FF      ;PREAMBLE TERM GROUP
1889                                              ;CHARACTER 2 = ALL ZEROS
1890  45F3  00                  .BYTE   $00      ;PREAMBLE TERM GROUP
1891                                              ;CHARACTER 3 = ALL ONES
1892  45F4  00                  .BYTE   $00      ;PREAMBLE TERM GROUP
1893                                              ;CHARACTER 4 = ALL ZEROS
1894  45F5  FF                  .BYTE   $FF      ;PREAMBLE TERM GROUP
1895                                              ;CHARACTER 5 = ALL ONES
1896
1897  45F6  FF                  .BYTE   $FF      ;PREAMBLE SECOND GROUP (01111)
1898                                              ;CHARACTER 1 = ALL ZEROS
1899  45F7  00                  .BYTE   $00      ;PREAMBLE SECOND GROUP
1900                                              ;CHARACTER 2 = ALL ONES
1901  45F8  FF                  .BYTE   $FF      ;PREAMBLE SECOND GROUP
1902                                              ;CHARACTER 3 = ALL ONES
1903  45F9  00                  .BYTE   $00      ;PREAMBLE SECOND GROUP
1904                                              ;CHARACTER 4 = ALL ONES
1905  45FA  FF                  .BYTE   $FF      ;PREAMBLE SECOND GROUP
1906                                              ;CHARACTER 5 = ALL ONES
1907                                              ;
1908  45FB  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 1 (14 GRPS OF 11111)
1909                                              ;CHARACTER 1 = ALL ONES
1910  45FC  FF                  .BYTE   $FF      ;PREAMBLE SYNC GROUP 1
1911                                              ;CHARACTER 2 = ALL ONES
1912  45FD  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 1
1913                                              ;CHARACTER 3 = ALL ONES
1914  45FE  FF                  .BYTE   $FF      ;PREAMBLE SYNC GROUP 1
1915                                              ;CHARACTER 4 - ALL ONES
1916  45FF  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 1
1917                                              ;CHARACTER 5 = ALL ONES
1918                                              ;
1919  4600  FF                  .BYTE   $FF      ;PREAMBLE SYNC GROUP 2
1920                                              ;CHARACTER 1 = ALL ONES
1921  4601  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 2
1922                                              ;CHARACTER 2 = ALL ONES
1923  4602  FF                  .BYTE   $FF      ;PREAMBLE SYNC GROUP 2
1924                                              ;CHARACTER 3 = ALL ONES
1925  4603  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 2
1926                                              ;CHARACTER 4 = ALL ONES
1927  4604  FF                  .BYTE   $FF      ;PREAMBLE SYNC GROUP 2
1928                                              ;CHARACTER 5 = ALL ONES
1929                                              ;
1930  4605  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 3
1931                                              ;CHARACTER 1 = ALL ONES
1932  4606  FF                  .BYTE   $FF      ;PREAMBLE SYNC GROUP 3
1933                                              ;CHARACTER 2 = ALL ONES
1934  4607  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 3
1935                                              ;CHARACTER 3 = ALL ONES
1936  4608  FF                  .BYTE   $FF      ;PREAMBLE SYNC GROUP 3
1937                                              ;CHARACTER 4 - ALL ONES
1938  4609  00                  .BYTE   $00      ;PREAMBLE SYNC GROUP 3
```

```
1939                                                    ;CHARACTER 5 = ALL ONES
1940                                                    ;
1941  460A  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 4
1942                                                    ;CHARACTER 1 = ALL ONES
1943  460B  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 4
1944                                                    ;CHARACTER 2 = ALL ONES
1945  460C  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 4
1946                                                    ;CHARACTER 3 = ALL ONES
1947  460D  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 4
1948                                                    ;CHARACTER 4 = ALL ONES
1949  460E  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 4
1950                                                    ;CHARACTER 5 = ALL ONES
1951                                                    ;
1952  460F  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 5
1953                                                    ;CHARACTER 1 = ALL ONES
1954  4610  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 5
1955                                                    ;CHARACTER 2 = ALL ONES
1956  4611  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 5
1957                                                    ;CHARACTER 3 = ALL ONES
1958  4612  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 5
1959                                                    ;CHARACTER 4 - ALL ONES
1960  4613  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 5
1961                                                    ;CHARACTER 5 = ALL ONES
1962                                                    ;
1963  4614  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 6
1964                                                    ;CHARACTER 1 = ALL ONES
1965  4615  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 6
1966                                                    ;CHARACTER 2 = ALL ONES
1967  4616  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 6
1968                                                    ;CHARACTER 3 = ALL ONES
1969  4617  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 6
1970                                                    ;CHARACTER 4 = ALL ONES
1971  4618  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 6
1972                                                    ;CHARACTER 5 = ALL ONES
1973                                                    ;
1974  4619  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 7
1975                                                    ;CHARACTER 1 = ALL ONES
1976  461A  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 7
1977                                                    ;CHARACTER 2 = ALL ONES
1978  461B  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 7
1979                                                    ;CHARACTER 3 = ALL ONES
1980  461C  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 7
1981                                                    ;CHARACTER 4 - ALL ONES
1982  461D  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 7
1983                                                    ;CHARACTER 5 = ALL ONES
1984                                                    ;
1985  461E  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 8
1986                                                    ;CHARACTER 1 = ALL ONES
1987  461F  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 8
1988                                                    ;CHARACTER 2 = ALL ONES
1989  4620  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 8
1990                                                    ;CHARACTER 3 = ALL ONES
1991  4621  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 8
1992                                                    ;CHARACTER 4 = ALL ONES
```

```
1993  4622  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 8
1994                                        ;CHARACTER 5 = ALL ONES
1995                                        ;
1996  4623  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 9
1997                                        ;CHARACTER 1 = ALL ONES
1998  4624  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 9
1999                                        ;CHARACTER 2 = ALL ONES
2000  4625  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 9
2001                                        ;CHARACTER 3 = ALL ONES
2002  4626  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 9
2003                                        ;CHARACTER 4 = ALL ONES
2004  4627  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 9
2005                                        ;CHARACTER 5 = ALL ONES
2006                                        ;
2007  4628  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 10
2008                                        ;CHARACTER 1 = ALL ONES
2009  4629  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 10
2010                                        ;CHARACTER 2 = ALL ONES
2011  462A  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 10
2012                                        ;CHARACTER 3 = ALL ONES
2013  462B  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 10
2014                                        ;CHARACTER 4 = ALL ONES
2015  462C  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 10
2016                                        ;CHARACTER 5 = ALL ONES
2017                                        ;
2018  462D  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 11
2019                                        ;CHARACTER 1 = ALL ONES
2020  462E  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 11
2021                                        ;CHARACTER 2 = ALL ONES
2022  462F  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 11
2023                                        ;CHARACTER 3 = ALL ONES
2024  4630  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 11
2025                                        ;CHARACTER 4 = ALL ONES
2026  4631  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 11
2027                                        ;CHARACTER 5 = ALL ONES
2028                                        ;
2029  4632  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 12
2030                                        ;CHARACTER 1 = ALL ONES
2031  4633  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 12
2032                                        ;CHARACTER 2 = ALL ONES
2033  4634  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 12
2034                                        ;CHARACTER 3 = ALL ONES
2035  4635  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 12
2036                                        ;CHARACTER 4 = ALL ONES
2037  4636  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 12
2038                                        ;CHARACTER 5 = ALL ONES
2039                                        ;
2040  4637  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 13
2041                                        ;CHARACTER 1 = ALL ONES
2042  4638  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 13
2043                                        ;CHARACTER 2 = ALL ONES
2044  4639  00          .BYTE   $00         ;PREAMBLE SYNC GROUP 13
2045                                        ;CHARACTER 3 = ALL ONES
2046  463A  FF          .BYTE   $FF         ;PREAMBLE SYNC GROUP 13
```

```
2047
2048  463B  00                        .BYTE   $00             ;CHARACTER 4 - ALL ONES
2049                                                          ;PREAMBLE SYNC GROUP 13
2050                                                          ;CHARACTER 5 = ALL ONES
2051  463C  FF                        .BYTE   $FF             ;
2052                                                          ;PREAMBLE SYNC GROUP 14
2053  463D  00                        .BYTE   $00             ;CHARACTER 1 = ALL ONES
2054                                                          ;PREAMBLE SYNC GROUP 14
2055  463E  FF                        .BYTE   $FF             ;CHARACTER 2 = ALL ONES
2056                                                          ;PREAMBLE SYNC GROUP 14
2057  463F  00                        .BYTE   $00             ;CHARACTER 3 = ALL ONES
2058                                                          ;PREAMBLE SYNC GROUP 14
2059  4640  FF                        .BYTE   $FF             ;CHARACTER 4 = ALL ONES
2060                                                          ;PREAMBLE SYNC GROUP 14
2061                                                          ;CHARACTER 5 = ALL ONES
2062  4641  FF                        .BYTE   $FF             ;
2063                                                          ;MARK ONE GROUP
2064  4642  FF                        .BYTE   $FF             ;CHARACTER 1 = ALL ZEROS
2065                                                          ;MARK ONE GROUP
2066  4643  00                        .BYTE   $00             ;CHARACTER 2 = ALL ZEROS
2067                                                          ;MARK ONE GROUP
2068  4644  FF                        .BYTE   $FF             ;CHARACTER 3 = ALL ONES
2069                                                          ;MARK ONE GROUP
2070  4645  00                        .BYTE   $00             ;CHARACTER 4 - ALL ONES
2071                                                          ;MARK ONE GROUP
2072                                                          ;CHARACTER 5 = ALL ONES
2073                                ;HERE IS THE DATA GROUP - 7 BYTES OF ALL 0'S (BYTE #86.)
2074                                ;ECC FOR DATA GRP = 0.
2075
2076  4646  FF                  GXPDAT: .BYTE   $FF             ;1ST CHAR IN 1ST HALF OF DATA GROUP - P - 0
2077  4647  00                          .BYTE   0               ;2ND CHAR - P = 1
2078  4648  00                          .BYTE   0               ;3RD CHAR - P = 1
2079  4649  00                          .BYTE   0               ;4TH CHAR - P = 1
2080  464A  FF                          .BYTE   $FF             ;5TH CHAR - P = 1
2081
2082  464B  00                          .BYTE   0               ;1ST CHAR IN 2ND HALF OF DATA GROUP - P = 0
2083  464C  FF                          .BYTE   $FF             ;2ND CHAR - P = 1
2084  464D  FF                          .BYTE   $FF             ;3RD CHAR - P = 1
2085  464E  FF                          .BYTE   $FF             ;4TH CHAR - P = 1
2086  464F  00                          .BYTE   0               ;5TH CHAR (ECC CHAR =0) - P = 1
2087
2088  4650  FF                  RESYNC: .BYTE   $FF             ;MARK TWO GROUP
2089  4651  00                          .BYTE   00
2090  4652  FF                          .BYTE   $FF
2091  4653  FF                          .BYTE   $FF
2092  4654  FF                          .BYTE   $FF
2093
2094  4655  00                          .BYTE   00              ;SYNC GROUP
2095  4656  FF                          .BYTE   $FF
2096  4657  00                          .BYTE   00
2097  4658  FF                          .BYTE   $FF
2098  4659  00                          .BYTE   00
2099
2100  465A  FF                          .BYTE   $FF             ;SYNC GROUP
```

```
2101  465B  00                        .BYTE   00
2102  465C  FF                        .BYTE   $FF
2103  465D  00                        .BYTE   00
2104  465E  FF                        .BYTE   $FF
2105
2106  465F  FF                        .BYTE   $FF             ;MARK ONE GROUP
2107  4660  FF                        .BYTE   $FF
2108  4661  00                        .BYTE   00
2109  4662  FF                        .BYTE   $FF
2110  4663  00                        .BYTE   00
2111
2112  4664  FF                        .BYTE   $FF             ;DATA SUB GROUP 1
2113  4665  00                        .BYTE   00
2114  4666  00                        .BYTE   00
2115  4667  00                        .BYTE   00
2116  4668  FF                        .BYTE   $FF
2117
2118  4669  00                        .BYTE   00              ;DATA SUB GROUP 2
2119  466A  FF                        .BYTE   $FF
2120  466B  FF                        .BYTE   $FF
2121  466C  FF                        .BYTE   $FF
2122  466D  00                        .BYTE   00
2123
2124                          ;END MARK = 11111 (ALL 1'S)
2125
2126  466E  FF                        .BYTE   $FF             ;END MARK GROUP
2127                                                          ;CHARACTER 1 = ALL ONES
2128  466F  00                        .BYTE   $00             ;END MARK GROUP
2129                                                          ;CHARACTER 2 = ALL ONES
2130  4670  FF                        .BYTE   $FF             ;END MARK GROPU
2131                                                          ;CHARACTER 3 = ALL ONES
2132  4671  00                        .BYTE   $00             ;END MARK GROUP
2133                                                          ;CHARACTER 4 = ALL ONES
2134  4672  FF                        .BYTE   $FF             ;END MARK GROUP
2135                                                          ;CHARACTER 5 = ALL ONES
2136                                                          ;
```

```
2138
2139                                    ;4 PAD BYTES
2140
2141  4673  00                                .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION - 1ST HALF
2142                                                                  ;CHARACTER 1
2143  4674  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2144                                                                  ;CHARACTER 2
2145  4675  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2146                                                                  ;CHARACTER 3
2147  4676  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2148                                                                  ;CHARACTER 4
2149  4677  00                                .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION
2150                                                                  ;CHARACTER 5
2151                                    ;2 PAD BYTES
2152                                    ;ACRC
2153                                    ;ECC
2154  4678  FF                                .BYTE   @377            ;RESIDUAL GROUP - AFTER TRANSLATION - 2ND HALF
2155                                                                  ;CHARACTER 1
2156  4679  82                                .BYTE   @202            ;RESIDUAL GROUP - AFTER TRANSLATION
2157                                                                  ;CHARACTER 2
2158  467A  82                                .BYTE   @202            ;RESIDUAL GROUP - AFTER TRANSLATION
2159                                                                  ;CHARACTER 3
2160  467B  24                                .BYTE   @044            ;RESIDUAL GROUP - AFTER TRANSLATION
2161                                                                  ;CHARACTER 4
2162  467C  59                                .BYTE   @131            ;RESIDUAL GROUP - AFTER TRANSLATION
2163                                                                  ;CHARACTER 5
2164                                    ;4 CRC BYTES
2165  467D  49                                .BYTE   @111            ;CRC GROUP - AFTER TRANSLATION - 1ST HALF
2166                                                                  ;CHARACTER 1
2167  467E  B6                                .BYTE   @266            ;CRC GROUP - AFTER TRANSLATION
2168                                                                  ;CHARACTER 2
2169  467F  59                                .BYTE   @131            ;CRC GROUP - AFTER TRANSLATION
2170                                                                  ;CHARACTER 3
2171  4680  B6                                .BYTE   @266            ;CRC GROUP - AFTER TRANSLATION
2172                                                                  ;CHARACTER 4
2173  4681  49                                .BYTE   @111            ;CRC GROUP - AFTER TRANSLATION
2174                                                                  ;CHARACTER 5
2175                                    ;2 CRC
2176                                    ;RESID
2177                                    ;ECC
2178  4682  3E                                .BYTE   @076            ;CRC GROUP - AFTER TRANSLATION - 2ND HALF
2179                                                                  ;CHARACTER 6
2180  4683  D1                                .BYTE   @321            ;CRC GROUP - AFTER TRANSLATION
2181                                                                  ;CHARACTER 7
2182  4684  3E                                .BYTE   @076            ;CRC GROUP - AFTER TRANSLATION
2183                                                                  ;CHARACTER 8
2184  4685  41                                .BYTE   @101            ;CRC GROUP - AFTER TRANSLATION
2185                                                                  ;CHARACTER 9
2186  4686  C9                                .BYTE   @311            ;CRC GROUP - AFTER TRANSLATION
2187                                                                  ;CHARACTER 10
2188                                                                  ;
```

```
2190
2191  4687  36                          .BYTE   @066        ;MARK TWO
                                                            ;CHARACTER 1 = ALL ONES
2192
2193  4688  C9                          .BYTE   @311        ;MARK TWO
                                                            ;CHARACTER 2 = ALL ONES
2194
2195  4689  36                          .BYTE   @066        ;MARK TWO
                                                            ;CHARACTER 3 = ALL ONES
2196
2197  468A  36                          .BYTE   @066        ;MARK TWO
                                                            ;CHARACTER 4 = ALL ZEROS
2198
2199  468B  36                          .BYTE   @066        ;MARK TWO
                                                            ;CHARACTER 5 = ALL ZEROS
2200
2201                                                        ;
2202  468C  C9                GDATEN:   .BYTE   @311        ;POSTAMBLE SYNC GROUP 1
                                                            ;CHARACTER 1 = ALL ONES
2203
2204  4     )  36                       .BYTE   @066        ;POSTAMBLE SYNC GROUP 1
                                                            ;CHARACTER 2 - ALL ONES
2205
2206  468E  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 1
                                                            ;CHARACTER 3 = ALL ONES
2207
2208  468F  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 1
                                                            ;CHARACTER 4 = ALL ONES
2209
2210  4690  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 1
                                                            ;CHARACTER 5 = ALL ONES
2211                                                        ;
2212
2213  4691  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 2
                                                            ;CHARACTER 1 = ALL ONES
2214
2215  4692  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 2
                                                            ;CHARACTER 2 - ALL ONES
2216
2217  4693  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 2
                                                            ;CHARACTER 3 = ALL ONES
2218
2219  4694  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 2
                                                            ;CHARACTER 4 = ALL ONES
2220
2221  4695  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 2
                                                            ;CHARACTER 5 = ALL ONES
2222                                                        ;
2223
2224  4696  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 3
                                                            ;CHARACTER 1 = ALL ONES
2225
2226  4697  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 3
                                                            ;CHARACTER 2 - ALL ONES
2227
2228  4698  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 3
                                                            ;CHARACTER 3 = ALL ONES
2229
2230  4699  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 3
                                                            ;CHARACTER 4 = ALL ONES
2231
2232  469A  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 3
                                                            ;CHARACTER 5 = ALL ONES
2233                                                        ;
2234
2235  469B  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 4
                                                            ;CHARACTER 1 = ALL ONES
2236
2237  469C  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 4
                                                            ;CHARACTER 2 - ALL ONES
2238
2239  469D  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 4
                                                            ;CHARACTER 3 = ALL ONES
2240
2241  469E  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 4
                                                            ;CHARACTER 4 = ALL ONES
2242
2243  469F  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 4
```

```
2244                                                    ;CHARACTER 5 = ALL ONES
2245                                                    ;
2246  46A0  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 5
2247                                                    ;CHARACTER 1 = ALL ONES
2248  46A1  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 5
2249                                                    ;CHARACTER 2 - ALL ONES
2250  46A2  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 5
2251                                                    ;CHARACTER 3 = ALL ONES
2252  46A3  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 5
2253                                                    ;CHARACTER 4 = ALL ONES
2254  46A4  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 5
2255                                                    ;CHARACTER 5 = ALL ONES
2256                                                    ;
2257  46A5  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 6
2258                                                    ;CHARACTER 1 = ALL ONES
2259  46A6  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 6
2260                                                    ;CHARACTER 2 - ALL ONES
2261  46A7  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 6
2262                                                    ;CHARACTER 3 = ALL ONES
2263  46A8  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 6
2264                                                    ;CHARACTER 4 = ALL ONES
2265  46A9  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 6
2266                                                    ;CHARACTER 5 = ALL ONES
2267                                                    ;
2268  46AA  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 7
2269                                                    ;CHARACTER 1 = ALL ONES
2270  46AB  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 7
2271                                                    ;CHARACTER 2 - ALL ONES
2272  46AC  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 7
2273                                                    ;CHARACTER 3 = ALL ONES
2274  46AD  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 7
2275                                                    ;CHARACTER 4 = ALL ONES
2276  46AE  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 7
2277                                                    ;CHARACTER 5 = ALL ONES
2278                                                    ;
2279  46AF  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 8
2280                                                    ;CHARACTER 1 = ALL ONES
2281  46B0  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 8
2282                                                    ;CHARACTER 2 - ALL ONES
2283  46B1  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 8
2284                                                    ;CHARACTER 3 = ALL ONES
2285  46B2  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 8
2286                                                    ;CHARACTER 4 = ALL ONES
2287  46B3  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 8
2288                                                    ;CHARACTER 5 = ALL ONES
2289                                                    ;
2290  46B4  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 9
2291                                                    ;CHARACTER 1 = ALL ONES
2292  46B5  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 9
2293                                                    ;CHARACTER 2 - ALL ONES
2294  46B6  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 9
2295                                                    ;CHARACTER 3 = ALL ONES
2296  46B7  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 9
2297                                                    ;CHARACTER 4 = ALL ONES
```

```
2298  46B8  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 9
2299                                         ;CHARACTER 5 = ALL ONES
2300                                         ;
2301  46B9  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 10
2302                                         ;CHARACTER 1 = ALL ONES
2303  46BA  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 10
2304                                         ;CHARACTER 2 - ALL ONES
2305  46BB  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 10
2306                                         ;CHARACTER 3 = ALL ONES
2307  46BC  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 10
2308                                         ;CHARACTER 4 = ALL ONES
2309  46BD  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 10
2310                                         ;CHARACTER 5 = ALL ONES
2311                                         ;
2312  46BE  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 11
2313                                         ;CHARACTER 1 = ALL ONES
2314  46BF  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 11
2315                                         ;CHARACTER 2 - ALL ONES
2316  46C0  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 11
2317                                         ;CHARACTER 3 = ALL ONES
2318  46C1  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 11
2319                                         ;CHARACTER 4 = ALL ONES
2320  46C2  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 11
2321                                         ;CHARACTER 5 = ALL ONES
2322                                         ;
2323  46C3  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 12
2324                                         ;CHARACTER 1 = ALL ONES
2325  46C4  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 12
2326                                         ;CHARACTER 2 - ALL ONES
2327  46C5  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 12
2328                                         ;CHARACTER 3 = ALL ONES
2329  46C6  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 12
2330                                         ;CHARACTER 4 = ALL ONES
2331  46C7  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 12
2332                                         ;CHARACTER 5 = ALL ONES
2333                                         ;
2334  46C8  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 13
2335                                         ;CHARACTER 1 = ALL ONES
2336  46C9  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 13
2337                                         ;CHARACTER 2 - ALL ONES
2338  46CA  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 13
2339                                         ;CHARACTER 3 = ALL ONES
2340  46CB  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 13
2341                                         ;CHARACTER 4 = ALL ONES
2342  46CC  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 13
2343                                         ;CHARACTER 5 = ALL ONES
2344                                         ;
2345  46CD  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 14
2346                                         ;CHARACTER 1 = ALL ONES
2347  46CE  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 14
2348                                         ;CHARACTER 2 - ALL ONES
2349  46CF  36          .BYTE   @066        ;POSTAMBLE SYNC GROUP 14
2350                                         ;CHARACTER 3 = ALL ONES
2351  46D0  C9          .BYTE   @311        ;POSTAMBLE SYNC GROUP 14
```

```
2352
2353  46D1  36                          .BYTE   @066    ;CHARACTER 4 = ALL ONES
2354                                                    ;POSTAMBLE SYNC GROUP 14
2355                                                    ;CHARACTER 5 = ALL ONES
2356  46D2  C9                          .BYTE   @311    ;
2357                                                    ;POSTAMBLE SECOND GROUP
2358  46D3  36                          .BYTE   @066    ;CHARACTER 1 = ALL ONES
2359                                                    ;POSTAMBLE SECOND GROUP
2360  46D4  C9                          .BYTE   @311    ;CHARACTER 2 = ALL ONES
2361                                                    ;POSTAMBLE SECOND GROUP
2362  46D5  36                          .BYTE   @066    ;CHARACTER 3 = ALL ONES
2363                                                    ;POSTAMBLE SECOND GROUP
2364  46D6  36                          .BYTE   @066    ;CHARACTER 4 = ALL ONES
2365                                                    ;POSTAMBLE SECOND GROUP
2366                                                    ;CHARACTER 5 = ALL ZEROS
2367  46D7  C9                          .BYTE   @311    ;
2368                                                    ;POSTAMBLE TERM GROUP
2369  46D8  C9                          .BYTE   @311    ;CHARACTER 1 = ALL ONES
2370                                                    ;POSTAMBLE TERM GROUP
2371  46D9  36                          .BYTE   @066    ;CHARACTER 2 = ALL ZEROS
2372                                                    ;POSTAMBLE TERM GROUP
2373  46DA  36                          .BYTE   @066    ;CHARACTER 3 = ALL ONES
2374                                                    ;POSTAMBLE TERM GROUP
2375  46DB  36              GTBLEN:     .BYTE   @066    ;CHARACTER 4 = ALL ZEROS
2376                                                    ;POSTAMBLE TERM GROUP
                                                        ;CHARACTER 5 = 0'S TO RESET TRACKS
```

```
                                        .SBTTL  TABLE EXPECTED PARITY

                                        ;HERE IS THE TABLE OF EXPECTED PARITY STATES FOR DATA & RESIDUAL GROUPS
                                        ;ALL BYTES IN DATA GRP=0 (EVEN ECC) SO 'P' TRACK 4 X 5 IS 0001 = 01111

46DC  00                        PXPTBL: .BYTE   0               ;1ST ENTRY IS A DUMMY
46DD  00                                .BYTE   0               ;'P' TRANSLATION ACTS SAME AS OTHER
46DE  01                                .BYTE   1               ;TRACKS
46DF  00                                .BYTE   0
46E0  01                                .BYTE   1
46E1  00                                .BYTE   0
46E2  00                                .BYTE   0
46E3  01                                .BYTE   1
46E4  00                                .BYTE   0
46E5  01                                .BYTE   1
46E6  00                                .BYTE   0

46E7  FF                                .BYTE   @377            ;SKIP OVER MARK TWO GROUP
46E8  FF                                .BYTE   @377
46E9  FF                                .BYTE   @377
46EA  FF                                .BYTE   @377
46EB  FF                                .BYTE   @377

46EC  FF                                .BYTE   @377            ;SKIP OVER SYNC GROUP
46ED  FF                                .BYTE   @377
46EE  FF                                .BYTE   @377
46EF  FF                                .BYTE   @377
46F0  FF                                .BYTE   @377

46F1  FF                                .BYTE   @377            ;SKIP OVER SYNC GROUP
46F2  FF                                .BYTE   @377
46F3  FF                                .BYTE   @377
46F4  FF                                .BYTE   @377
46F5  FF                                .BYTE   @377

46F6  FF                                .BYTE   @377            ;SKIP OVER MARK ONE GROUP
46F7  FF                                .BYTE   @377
46F8  FF                                .BYTE   @377
46F9  FF                                .BYTE   @377
46FA  FF                                .BYTE   @377

46FB  00                                .BYTE   0               ;PTRACK FOR DATA GROUP
46FC  01                                .BYTE   1
46FD  00                                .BYTE   0
46FE  01                                .BYTE   1
46FF  00                                .BYTE   0
4700  00                                .BYTE   0
4701  01                                .BYTE   1
4702  00                                .BYTE   0
4703  01                                .BYTE   1
4704  00                                .BYTE   0

                                ;SKIP OVER END MARK GRP
```

```
 2432  4705  FF                          .BYTE   @377
 2433  4706  FF                          .BYTE   @377
 2434  4707  FF                          .BYTE   @377
 2435  4708  FF                          .BYTE   @377
 2436  4709  FF                          .BYTE   @377
 2437
 2438                              ;RESIDUAL GRP
 2439                              ;1ST GRP IS 4 PADS
 2440
 2441  470A  01                          .BYTE   1
 2442  470B  00                          .BYTE   0
 2443  470C  01                          .BYTE   1
 2444  470D  00                          .BYTE   0
 2445  470E  01                          .BYTE   1
```

```
                                        ;2ND GRP IS 2 PAD BYTES + ACRC OF 323 + ECC OF 225

470F  01                                        .BYTE   1
4710  00                                        .BYTE   0
4711  01                                        .BYTE   1
4712  00                                        .BYTE   0
4713  01                                        .BYTE   1

                                        ;CRC GRP 1ST HALF IS 4 CRC BYTES OF 136

4714  00                                        .BYTE   0
4715  01                                        .BYTE   1
4716  01                                        .BYTE   1
4717  01                                        .BYTE   1
4718  00                                        .BYTE   0

                                        ;2ND HALF OF CRC GRP IS 2 CRC OF 136 + RESID BYTE OF 006 + ECC BYTE OF 352

4719  01                                        .BYTE   1
471A  01                                        .BYTE   1
471B  01                                        .BYTE   1
471C  00                                        .BYTE   0
471D  01                                        .BYTE   1

                                        ;HERE IS THE MARK-2 PARITY TABLE

471E  00                                        .BYTE   0               ;MARK 2 IS 11100
471F  01                                        .BYTE   1
4720  00                                        .BYTE   0
4721  00                                        .BYTE   0
4722  00                                        .BYTE   0
```

```
2480                                    .SBTTL  PROGRAM VARIABLES
2481
2482  4723   00                         CCTLWD: .BYTE   0               ;CLOCK CONTROL WORD
2483  4724   00                         GRPCNT: .BYTE   0               ;GROUP COUNTER
2484  4725   00                         UNITMP: .BYTE   0               ;UNIT MAP
2485  4726   00                         XDATA:  .BYTE   0               ;COPY OF EXPECTED DATA
2486  4727   00   00                    DPCNT:  .WORD   0               ;PARITY TEST COUNTER
2487  4729   00   00                    EXDLPC: .WORD   0               ;TRANSITION COUNT AT ERROR
2488  472B   00   00                    POINTER: .WORD  0               ;POINTER TO EXPECTED DATA TABLE
2489  472D   00   00                    TRANSCNT: .WORD 0               ;TRANSITION COUNT
2490
2491        0000                              .END
```

```
A     =%0007    ABOCNO  43FD  G   ADATA = 0094    AMTIEP= 0001
AMTIE7= 0002    ARAIDF= 0098      ASAVE   4F9B    ATTCD   4F97
AXNUM   4F91    B     =%0000      BADST = 0090    BGNPRE  4406
BIT0  = 0001    BIT1  = 0002      BIT15 = 8000    BIT2  = 0004
BIT3  = 0008    BIT4  = 0010      BIT5  = 0020    BIT6  = 0040
BIT7  = 0080    BIT8  = 0100      BIT9  = 0200    BRKPBC= 4F0A
BRKRAM= 4F10    BRKSTR= 4E60      BRKXCT= 4F00    BSAVE   4F9C
BYTCNT= 00D4    BYTEH   4F24      BYTEL   4F23    C     =%0001
CASCT   4F21    CASCTL= 00A0      CASSTA= 00A0    CATTH = 0089
CATTL = 0088    CBUSST= 00A1      CBYTH = 008B    CBYTL = 008A
CCTLWD  4723    CDG1H = 0087      CDG1L = 0086    CDG2H = 0093
CDG2L = 0092    CDG3H = 0095      CDG3L = 0094    CDVTH = 008D
CDVTL = 008C    CHPTIE= 0028      CH0TIE= 0020    CH1TIE= 0021
CH2TIE= 0022    CH3TIE= 0023      CH4TIE= 0024    CH5TIE= 0025
CH6TIE= 0026    CH7TIE= 0027      CKLOP = 2815    CLEAR   45C6
CLK     43EA    CLKCTL= 00F0      CLKRS   43DD    CLKSYS  45BA
CLK1    43E8    CLOCK   4F26      CLRLP   45CA    CMC0H = 0099
CMC0L = 0098    CMC1H = 009B      CMC1L = 009A    CMC2H = 009D
CMC2L = 009C    CMC3H = 009F      CMC3L = 009E    CMINH = 0097
CMINL = 0096    CNTCTL= 00D7      CONT    4529    CRCWRD= 0018
CSAVE   4F9D    CSRLH = 0091      CSRLL = 0090    CTCH  = 0085
CTCL  = 0084    CTSTH = 008F      CTSTL = 008E    CXCTH = 0081
CXCTL = 0080    CXINH = 0083      CXINL = 0082    C.    = 0001
C.AVAI= 0080    C.DP  = 0008      C.DSE = 0010    C.DTU = 0003
C.DVA = 0008    C.FAIL= 00FC      C.FMT = 0070    C.FNCT= 003E
C.GO  = 0001    C.INTC= 00FE      C.MAIN= 0020    C.NSA = 0080
C.RCT = 00FC    C.SER = 0080      C.SHR = 0040    C.SKPC= 000F
C.TAPE= 0040    C.WCS = 0002      D     =%0002    DATACT= 00D0
DBUS    4F28    DBUSCT= 00C0      DBUSST= 00C0    DDRA  = 00D8
DDRAIN= 0010    DDRB  = 00D9      DDRBIN= 0002    DDRC  = 00DA
DDRCIN= 0001    DDRCO = 0088      DDRCTL= 00DB    DIAFLG  4F22
DIAGPG= 4300    DIAGRM= 4F90      DIARD = 000B    DONE1 = 0045
DONINT= 0010    DPCNT   4727      DPTST   4498    DSAVE   4F9E
DSE   = 0006    DUMMY   4374  G   D.ATH0= 0001    D.ATH1= 0002
D.EOTD= 0010    D.LAGC= 0020      D.NOTW= 0040    D.NTHR= 0004
D.TACH= 0008    D.WR4 = 0080      E     =%0003    ECCBAD= 0042
ECCCOR= 0019    ECCOK = 0041      ECCSTA= 001A    ECCTST= 000E
EDATA = 0095    EOTCLR= 0003      ERFLG   4F93    ERLP  = 2809
ERLPA = 280F    ERLPB = 2812      ERLPE = 280C    ERNUM   4F90
ERRCNT= 00D6    ESAVE   4F9F      EXDLPC  4729    EXIT    45B7
E.ACRC= 0010    E.AMT = 0020      E.CDP = 0080    E.CRC = 0080
E.PNTR= 0008    E.RPE = 0040      E.STEC= 0001    E.TTEC= 0002
E.UNC = 0004    FIFORD= 006A      FORMAT  4F25    FOUND   437D
FWDTST= 0061    GCRID = 0089      GCRSET= 0002    GDATEN  468C
GOODTM= 0092    GRPCNT  4724      GTBLEN  46DB    GXPDAT  4646
GXPTBL  45F1    H     =%0004      HLSAVE  4FA0    IE    = 0008
INTSTA= 00E0    ITERA   4F9A      I.PWR = 0020    I.RMPE= 0040
I5.5  = 0010    I6.5  = 0020      I7.5  = 0040    KCALL = 005F
KCLR  = 007B    KDEP  = 003F      KENAB = 0078    KEXAM = 003E
KEYBRD= 00C8    KEY1  = 0078      KEY10 = 006D    KEY11 = 006E
KEY12 = 006F    KEY13 = 005C      KEY14 = 005D    KEY15 = 005E
KEY16 = 005F    KEY17 = 003C      KEY18 = 003D    KEY19 = 003E
KEY2  = 0079    KEY20 = 003F      KEY3  = 007A    KEY4  = 007B
KEY5  = 0074    KEY6  = 0075      KEY7  = 0076    KEY8  = 0077
```

```
KEY9   = 006C          KINTA = 006F          KLDAD = 003D          KN0    = 003C
KN1    = 005C          KN2   = 005D          KN3   = 005E          KN4    = 006C
KN5    = 006D          KN6   = 006E          KN7   = 0074          KN8    = 0075
KN9    = 0076          KU2   = 0079          KU3   = 007A          KU8    = 0077
L      =%0005          LBLANK= 000F          LCE   = 000B          LCH    = 000C
LCL    = 000D          LCP   = 000E          LC0   = 0000          LC1    = 0001
LC2    = 0002          LC3   = 0003          LC4   = 0004          LC5    = 0005
LC6    = 0006          LC7   = 0007          LC8   = 0008          LC9    = 0009
LDLEDA= 00CA           LDLEDB= 00CB          LDLEDC= 00CC          LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF          LKDIAG= 2800          LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058          LKLWMP= 0055          LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046          LKOPR = 0046          LPFLG    4F94
LPNUM    4F92          M      =%0006         MBSEL = 00E0          MB.A   = 0008
MB.B   = 0004          MEMTOP= 4FFF          MINUS = 000A          MM     = 8000
MSE    = 0008          MSGN  = 0008          MTACLR= 0000          MT.ARA= 0020
MT.CPE= 0080           MT.DSE= 0001          MT.FWD= 0040          MT.INH= 0008
MT.LWR= 0004           MT.MOT= 0002          MT.NWT= 0080          MT.PEC= 0040
MT.PSB= 0004           MT.PS0= 0001          MT.PS1= 0002          MT.REV= 0020
MT.WRT= 0010           MT.Z  = 0008          M.ATA = 0080          M.CAPE= 0020
M.CONT= C080           M.DEM = 0020          M.EBL = 0010          M.EXC  = 0008
M.FAIL= 0008           M.ILR = 0010          M.INIT= 0010          M.OCC  = 0020
M.ONLI= 0001           M.PE  = 0040          M.PORT= 0080          M.RDEN= 0002
M.RDPE= 0008           M.RUN = 0004          M.SCLK= 0001          M.TRA  = 0040
M.UNIT= 0007           M.WCLK= 0040          M.WCLN= 0080          M.WREN= 0080
M5.5   = 0001          M6.5  = 0002          M7.5  = 0004          NOTCAP= 0088
OKAY   = 00FF          OPRRAM= 4300          OPSTRT= 0058          OPVER  = 0040
PADCNT= 00D5           PADCRC= 0080          PDIAG = 0048          PEID   = 008A
PENAB  = 004C          PESET = 0001          PL    = 00B1          POINTE   472B
PPOST    44AB          PPREAM   44AB         PRDD  = 004C          PRECN1   4441  G
PRECN3   453E  G       PRENF = 009C          PS    = 00B2          PSTAT  = 0048
PSW    =%0009          PTSTOK   44CD  G      PTY      44B0         PTYC     44F8
PTY1     44BB          PTY2     44C8         PXPTBL   46DC         P.AMTP= 0001
P.BCTC= 0040           P.CMDP= 0020          P.INTE= 0080          P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002         P.RPEN= 0004          P.RPST= 0002
P.RPOE= 0020           P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010
P.SING= 0080           P.STAT= 0002          P.STPE= 0080          P.TACH= 0008
P.TUPR= 0010           P.WCSP= 0004          P.WDS  = 0040         P.WFLP= 0001
P.WPEN= 0010           P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008
P.WP3E= 0004           P.5VOK= 0002          QUE    = 281B         QUEM   = 281E
RAMT   = 0010          RARA  = 0006          RARAI = 0004          RCHBDO= 0048
RCHBD1= 0047           RCHOK = 0046          RCHTST= 000C          RCLRT  = 000D
RCMD   = 0008          RCMLP = 0003          RCONT = 0080          RDATA  = 0017
RDCLK  = 0010          RDON  = 0011          READG = 0007          RECCN1   4574  G
RECCN2   459D  G       RECCN3   458F  G      RECEND   4558         REND   = 0014
REQST  = 2806          RESCHR= 00D1          RESYNC   4650         REVTST= 0064
REWIND= 0004           RFIFOL= 0008          RGCLK = 0002          RGCRI  = 0003
RIBG   = 0001          RILL  = 0012          RINST = 000C          RMCTST= 0008
RMK2   = 0013          RNOP  = 0000          RPATH = 0001          RPBAD  = 0044
RPCHI  = 0001          RPCLK = 0003          RPCTL = 0009          RPEI   = 0002
RPFAIL= 0000           RPF1  = 009D          RPF2  = 009E          RPOK   = 0043
RPOSTN= 0016           RPSTA = 0015          RRCMT = 000A          RSTAT  = 0002
RTIEB  = 000A          RTIER = 0030          RTM    = 0005         RUNKI  = 0009
RUPTST= 005E           RWDUNL= 0005          R.AMT = 0001          R.BOP  = 0008
R.DATA= 0040           R.DON  = 0002         R.DRDY= 0010          R.END  = 0010
```

XMC6 - TRANSLATOR MICRO CONTROLLER PART #6 CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:11 PAGE 1-33 SEQ 0615
XMC6.M80 SYMBOL TABLE

```
R.ILL  = 0004          R.JVOK= 0006          R.MK2  = 0008          R.PLOD= 0008
R.PLO0= 0010          R.PLO1= 0020          R.POST= 0020          R.STNM= 0002
R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040
R.VOK  = 0080          R00H   = 0081          R00L   = 0080          R01H   = 0083
R01L   = 0082          R02H   = 0085          R02L   = 0084          R03H   = 0087
R03L   = 0086          R04H   = 0089          R04L   = 0088          R05H   = 008B
R05L   = 008A          R06H   = 008D          R06L   = 008C          R07H   = 008F
R07L   = 008E          R10H   = 0091          R10L   = 0090          R11H   = 0093
R11L   = 0092          R12H   = 0095          R12L   = 0094          R13H   = 0097
R13L   = 0096          R14H   = 0099          R14L   = 0098          R15H   = 009B
R15L   = 009A          R16H   = 009D          R16L   = 009C          R17H   = 009F
R17L   = 009E          R7.5   = 0010          SELCLR= 0000          SEQERR  4598
SETATA= 00A1          SID    = 0080          SOD    = 0080          SOE    = 0040
SP     =%0008          SSCLK  = 0040          SSTEP  = 0005          STACK  = 4FFF
STATRM= 4F20          STPCT   4F20          STRSP  = 5000          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT   = 0044
TASEL  = 0080          TCMD   = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST1   4300          TMF    = 0099          TMRDY  = 0040          TRANSC  472D
TRKENA= 00D2          TSET   = 2803          TSTEND= 2818          TSTS   = 0040
TUSEL0= 00D1          TUSEL1= 00D2          TU78   = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT  = 0004          T.EOT  = 0002          T.FPT  = 0001
T.NTHR= 0004          T.ONL  = 0020          T.PES  = 0008          T.PSBJ= 0020
T.PSOJ= 0008          T.PS1J= 0010          T.RDY  = 0080          T.RDY0= 0040
T.RWD  = 0010          T.SCLK= 0002          T1LOOP  431E          T1LPA   434C
UIBG   = 00A1          UNITMP  4725          VALFC   4F98          VALTB   4F95
VELTST= 005B          WDR.P  = 0010          WMCCTL= 00D3          WMCERR= 00DA
WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004
W.CRC  = 0008          W.DIAG= 0002          W.DONN= 0040          W.ECC  = 0010
W.ENAB= 0080          W.ERR  = 0020          W.FMT  = 0070          W.GCR  = 0010
W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002          W.REV  = 0004
W.ROME= 0010          W.RST  = 0001          W.SKIP= 000F          W.WRIT= 0008
W.XFER= 0020          X      =%000A          XDATA   4726          XDLOOP  444A
XDLPC   446C          XDLPC1  442F          XDLPC2  443C          XDLP1   445A
XLLP1   4305          XMEND   45A3          XPLOOP  .4D6          XPLP1   44E6
X.DONN- 0080          X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001
X.WCLK= 0001          Y      =%000B          .      = 472F
```

ERRORS DETECTED: 0

*XMC6.A78/PTP,XMC6=NLIST,PARAM,MACRO,LIST,XMC6
RUN-TIME: 4 5 0 SECONDS
CORE USED: 10K

```
1329                                    .TITLE  RPM1 - READ PATH MICROCONTROLLER PART #1
1330                                    .SBTTL  TEST 01 - TIE BUS INTEGRITY TEST
1331                                    ;ID     RPM1-READ PATH MICRO CONTROLLER PART #1
1332
1333  4300                              ST
 (1)                                    ; **********************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1334                                    ;*RPM - TIE BUS INTEGRITY TEST
1335  4300                              SD
 (1)                                    ; **********************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1336                                    ;*THIS TEST CHECKS FOR STUCK BITS ON THE TIE BUS BY WRITING 16 (4 BIT)
1337                                    ;*DATA PATTERNS (0-17) TO THE TIE BUS AND READING THEM FROM THE TIE BUS.
1338  4300                              SP
 (1)                                    ; **********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1339                                    ;*BGNTST
1340                                    ;*  SET NORMAL READ PATH CLOCKS
1341                                    ;*  CLEAR TIE BUS VALUE
1342                                    ;*  SET THE JAM TIE BUS BIT
1343                                    ;*  BGNDO
1344                                    ;*  :  OUTPUT TIE BUS VALUE TO THE TIE BUS
1345                                    ;*  :  WAIT (NOP)
1346                                    ;*  :  INPUT THE TIE BUS VALUE
1347                                    ;*  :  IF TIE BUS OUTPUT=TIE BUS INPUT
1348                                    ;*  :  :  THEN-CONTINUE
1349                                    ;*  :  :  ELSE-ERROR
1350                                    ;*  :  ENDIF
1351                                    ;*  :  INCREMENT THE COMMAND VALUE
1352                                    ;*  :  DO UNTIL THE COMMAND VALUE=20(8)
1353                                    ;*  ENDDO
1354                                    ;*ENDTST
1355  4300                              SE
 (1)                                    ; **********************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1356                                    ;*RPM1 MICRO TEST 01
1357                                    ;*RPM1 MICRO ERROR 01
1358                                    ;*RPM1-READ PATH-TIE BUS INTEGRITY TEST
1359                                    ;*M8953, M8950, M8951
1360                                    ;*TIE DATA WRITTEN NOT=READ
1361                                    ;*ACTUAL = NNNN
1362                                    ;*EXPECTED = NNNN
1363  4300                              S
 (1)                                    ; **********************************************************************
1364  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01              7.0                     MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1365                                    ;%RPM1-READ PATH-TIE BUS INTEGRITY TEST
1366                                    ;&M8953, M8950, M8951, M8952
1367
```

```
1368  4305  3E  10          7.0  TST01X: MVI   A,RDCLK          ;SET NORMAL READ PATH CLOCKS
1369  4307  D3  F0         10.0          OUT   CLKCTL
1370
1371  4309  AF             4.0          XRA   A                ;CLEAR TIE BUS JAM VALUE
1372  430A  32  49  4B     13.0          STA   JAMDAT
1373
1374  430D  3E  80           7.0          MVI   A,R.TBJN                ;SET JAM TIE BUS BIT
1375  430F  D3  09          10.0          OUT   RPCTL
1376
1377  4311  3A  49  4B      13.0  TIELO:  LDA   JAMDAT           ;OUTPUT TIE BUS DATA TO TIE BUS
1378  4314  D3  0A          10.0          OUT   RTIEB
1379  4316                        ROUT  EDATA
 (1)  4316  D3  95          10.0                OUT   EDATA             ;WRITE AC INTO EDATA
 (1)  4318  7F              4.0                 MOV   A,A               ;RETRY LINK
1380  4319  47              4.0          MOV   B,A              ;SAVE TIE BUS WRITTEN IN B
1381  431A  00              4.0          NOP                    ;WAIT
1382  431B  DB  30          10.0          IN    RTIER            ;INPUT THE TIE BUS VALUE
1383  431D  B8              4.0          CMP   B                ;IF TIE BUS OUTPUT=TIE BUS INPUT
1384  431E  CA  29  43      10.0          JZ    TIECNO           ;THEN CONTINUE
1385  4321                        ROUT  ADATA                  ;ELSE-ERROR
 (1)  4321  D3  94          10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  4323  7F              4.0                 MOV   A,A               ;RETRY LINK
1386  4324                        ERRB  TIELO,TIECNO
 (1)                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4324  CD  12  28      18.0                CALL  ERLPB             ;PROCESS ERROR - DO 2.3
 (1)        0001                              MSGN  =     MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4327  01                                .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  4328  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4329  CD  15  28      18.0          TIECNO::              CALL  CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  432C  DA  11  43      10.0                JC    TIELO            ;LOOP ADDRESS IF LOOP SPECIFIED
1387                              :>TIE DATA WRITTEN NOT=READ
1388
1389  432F  3A  49  4B      13.0          LDA   JAMDAT
1390  4332  3C              4.0          INR   A
1391  4333  32  49  4B      13.0          STA   JAMDAT
1392  4336  FE  10          7.0          CPI   @20
1393  4338  C2  11  43      10.0          JNZ   TIELO
1394  433B                        ENDTST  TST01X
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  433B                        REQ   7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  433B  CD  06  28      18.0                          CALL    REQST
 (2)  433E  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  433F  00  00                                        .WORD                   ;SYSTEM '*' COUNT
 (2)  4341  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4343  00                                            .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4344  07                                            .BYTE   7               ;REQUEST CODE
 (1)  4345  3A  9A  4F      13.0          LDA   ITERA            ;GET ITERATION COUNT
 (1)  4348  3D              4.0          DCR   A                ;DOWNCOUNT
 (1)  4349  32  9A  4F      13.0          STA   ITERA            ;SAVE COUNT
 (1)  434C  F2  05  43      10.0          JP    TST01X           ;DO TEST UNTIL TILL = 0
1395
```

```
                                .SBTTL  TEST 02 - LOOP COMMAND WRITE TO COMMAND READ

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM - LOOP COMMAND WRITE TO COMMAND READ
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE READ PATH COMMAND REGISTER BY WRITING 256 DATA
                                ;*PATTERNS 000-377(8) TO THE READ PATH COMMAND REGISTER AND CHECKING
                                ;*THE RESULTS AT THE LOOP AROUND COMMAND REGISTER.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CLEAR THE COMMAND VALUE TO ZERO
                                ;*    ENABLE THE READ MICRO CONTROLLER
                                ;*    BGNDO
                                ;*       :    OUTPUT COMMAND VALUE TO READ PATH COMMAND REGISTER
                                ;*       :    INPUT COMMAND FROM READ PATH COMMAND REGISTER
                                ;*       :        IF COMMAND OUTPUT = COMMAND INPUT
                                ;*       :            THEN-CONTINUE
                                ;*       :            ELSE-ERROR
                                ;*       :        ENDIF
                                ;*       :    INCREMENT THE COMMAND VALUE
                                ;*       :    DO UNTIL COMMAND VALUE = ZERO
                                ;*    ENDDO
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 02
                                ;*RPM1 MICRO ERROR 02
                                ;*RPM1-READ PATH-LOOP COMMAND WRITE TO READ
                                ;*M8953, M8960
                                ;*DATA WRITTEN NOT = READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                S
                                ; ************************************************************************

                                TEST2:  TESTX   @2              ;INITIALIZE THE TEST
                                                MVI     A,@2            ;DEFINE THE TEST NUMBER
                                                CALL    TSET            ;SETUP THE TEST
                                ;%RPM1-READ PATH-LOOP COMMAND WRITE TO READ
                                ;&M8953, M8960, M8951, M8952

                                TST02X: MVI     A,RDCLK
```

```
1436  4356  D3  F0          10.0          OUT     CLKCTL          ;SET NORMAL CLOCKS
1437                                                              ;
1438  4358  AF               4.0          XRA     A
1439  4359  32  46  4B      13.0          STA     DATAE           ;CLEAR THE COMMAND VALUE
1440                                                              ;
1441  435C  3E  00           7.0          MVI     A,0             ;LOAD THE READ PATH ENABLE BIT
1442  435E  D3  09          10.0          OUT     RPCTL           ;ENABLE THE READ PATH
1443
1444  4360  21  46  4B      10.0  RMCLP0: LXI     H,DATAE         ;LOAD ADDRESS OF THE DATA BYTE
1445  4363  7E               7.0          MOV     A,M             ;GET THE DATA BYTE
1446  4364  D3  08          10.0          OUT     RCMD            ;OUTPUT TO THE READ MICRO
1447                                                              ;
1448  4366  DB  03          10.0          IN      RCMLP           ;INPUT FROM THE READ MICRO
1449  4368  BE               7.0          CMP     M               ;EQUAL?
1450  4369  CA  78  43      10.0          JZ      RMCCN0
1451  436C                                ROUT    ADATA           ;NO-STORE ACTUAL DATA
 (1)  436C  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  436E  7F               4.0                  MOV     A,A             ;RETRY LINK
1452  436F  7E               7.0          MOV     A,M             ;GET THE EXPECTED DATA
1453  4370                                ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  4370  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4372  7F               4.0                  MOV     A,A             ;RETRY LINK
1454  4373                                ERRB    RMCLP0,RMCCN0
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4373  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0002                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4376  02                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4377  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4378  CD  15  28      18.0          RMCCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  437B  DA  60  43      10.0                  JC      RMCLP0          ;LOOP ADDRESS IF LOOP SPECIFIED
1455                            ;>DATA WRITTEN NOT = READ
1456  437E  3A  46  4B      13.0          LDA     DATAE           ;GET THE DATA BYTE
1457  4381  3C               4.0          INR     A               ;INCREMENT
1458  4382  32  46  4B      13.0          STA     DATAE           ;STORE THE DATA BYTE
1459  4385  A7               4.0          ANA     A               ;SET THE CONDITION CODE
1460  4386  C2  60  43      10.0          JNZ     RMCLP0          ;DONE?
1461  4389                                ENDTST  TST02X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4389                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4389  CD  06  28      18.0                          CALL    REQST
 (2)  438C  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  438D  00  00                                        .WORD                   ;SYSTEM '*' COUNT
 (2)  438F  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4391  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4392  07                                            .BYTE   7               ;REQUEST CODE
 (1)  4393  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4396  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4397  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  439A  F2  54  43      10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1462
```

```
                                .SBTTL  TEST 03 - STATUS REGISTER TEST

                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM - STATUS REGISTER TEST
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS EACH BIT IN THE READ PATH MICRO CONTROLLER STATUS BYTE
                                ;*BY ISSUING THE COMMANDS 012 - 172 TO THE READ PATH. THESE TEST COMMANDS
                                ;*LOAD THE STATUS REGISTER AS FOLLOWS:
                                ;*
                                ;*      COMMAND         STATUS
                                ;*      -------         ------
                                ;*
                                ;*       012              001
                                ;*       032              002
                                ;*       052              004
                                ;*       072              010
                                ;*       112              020
                                ;*       132              040
                                ;*       152              100
                                ;*       172              200
                                ;*
                                ;*AFTER EACH COMMAND IS ISSUED THE RESULTING STATUS IS VERIFIED.
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*       BGNDO
                                ;*        ENABLE THE READ PATH
                                ;*        SET EXPECTED STATUS BYTE = 1
                                ;*        LOAD THE COMMAND BYTE WITH 12(8)
                                ;*        BGNDO
                                ;*        :   OUTPUT THE COMMAND BYTE
                                ;*        :   DELAY 5 MICRO SECONDS
                                ;*        :   INPUT THE READ PATH ACTUAL STATUS BYTE
                                ;*        :      IF INPUT STATUS BYTE = EXPECTED STATUS BYTE
                                ;*        :          THEN-CONTINUE
                                ;*        :          ELSE-ERROR
                                ;*        :      ENDIF
                                ;*        :    SHIFT THE EXPECTED STATUS LEFT ONE BIT
                                ;*        :    ADD 20(8) THE COMMAND BYTE
                                ;*        :    DO UNTIL THE COMMAND BYTE = 212(8)
                                ;*     ENDDO
                                ;*     DO UNTIL ALL CLOCKS TESTED
                                ;*    ENDDO
                                ;*ENDTST
                                SE
```

```
 (1)                                    ; ************************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1509                                    ;*RPM1 MICRO TEST 03
1510                                    ;*RPM1 MICRO ERROR 03
1511                                    ;*RPM1-READ PATH-STATUS BYTE TEST
1512                                    ;*M8953, M8960
1513                                    ;*ACTUAL STATUS NOT = EXPECTED STATUS
1514                                    ;*ACTUAL = NNNN
1515                                    ;*EXPECTED = NNNN
1516  439D                              S
 (1)                                    ; ************************************************************************************
1517
1518  439D                              TEST3:  TESTX   @3              ;INITIALIZE THE TEST
 (1)  439D  3E  03            7.0                MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  439F  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1519                                    ;%RPM1-READ PATH-STATUS BYTE TEST
1520                                    ;&M8953, M8960
1521
1522  43A2  A7                  4.0   TST03X: ANA     A
1523  43A3  3E  10              7.0   TST03A: MVI     A,RDCLK
1524  43A5  D3  F0             10.0           OUT     CLKCTL          ;SET NORMAL CLOCKS
1525  43A7  CD  12  44          18.0           CALL    STATCK          ;GO DO STATUS BYTE TEST
1526  43AA  D2  B2  43          10.0           JNC     TST03B          ;NO ERRORS - GO USE NEXT CLOCK RATE
1527  43AD                              ERRB    TST03A,TST03B
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43AD  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0003                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43B0  03                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43B1  00                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43B2  CD  15  28       18.0   TST03B::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43B5  DA  A3  43       10.0                JC      TST03A          ;LOOP ADDRESS IF LOOP SPECIFIED
1528                            ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = NORMAL(20.00 MHZ)
1529                                    ;
1530  43B8  A7                  4.0           ANA     A
1531  43B9  3E  00              7.0           MVI     A,0             ;
1532  43BB  D3  F0             10.0           OUT     CLKCTL          ;SET -10% CLOCK
1533  43BD  CD  12  44          18.0           CALL    STATCK          ;GO DO STATUS BYTE TEST
1534  43C0  D2  C8  43          10.0           JNC     TST03C          ;NO ERRORS - GO DO THE NEXT CLOCK RATE
1535  43C3                              ERRB    TST03B,TST03C
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43C3  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43C6  04                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43C7  00                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43C8  CD  15  28       18.0   TST03C::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43CB  DA  B2  43       10.0                JC      TST03B          ;LOOP ADDRESS IF LOOP SPECIFIED
1536                            ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = -10%(18.08 MHZ)
1537                                    ;
1538  43CE  A7                  4.0           ANA     A
1539  43CF  3E  08              7.0           MVI     A,@10           ;
1540  43D1  D3  F0             10.0           OUT     CLKCTL          ;SET +10% CLOCK
1541  43D3  CD  12  44          18.0           CALL    STATCK  ;GO DO STATUS BYTE TEST
1542  43D6  D2  DE  43          10.0           JNC     TST03D          ;NO ERRORS - GO USE NEXT CLOCK RATE
```

```
                                        ERRB    TST03C,TST03D
                                        ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE                   ;PRINT ROUTINE NUMBER
                                        TST03D::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      TST03C          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = +10%(21.84 MHZ)
                                                ;
 43E4  A7                4.0            ANA     A
 43E5  3E  18            7.0            MVI     A,030           ;
 43E7  D3  F0           10.0            OUT     CLKCTL          ;SET -30% CLOCK
 43E9  CD  12  44       18.0            CALL    STATCK          ;GO DO STATUS BYTE TEST
 43EC  D2  F4  43       10.0            JNC     TST03E          ;NO ERRORS - WERE DONE - GO HOME
 43EF                                   ERRB    TST03D,TST03E
                                        ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 43EF  CD  12  28       18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
       0006                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 43F2  06                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 43F3  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 43F4  CD  15  28       18.0            TST03E::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 43F7  DA  DE  43       10.0                    JC      TST03D          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = -30%(15.60 MHZ)
                                                ;
 43FA  A7                4.0            ANA     A
 43FB                                   ENDTST  TST03X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 43FB                                           CALL    REQST
 43FB  CD  06  28       18.0                    .BYTE                   ;DATA PATTERN NUMBER
 43FE  00                                       .WORD                   ;SYSTEM '*' COUNT
 43FF  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 4401  00  00                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 4403  00                                       .BYTE   7               ;REQUEST CODE
 4404  07
 4405  3A  9A  4F       13.0            LDA     ITERA           ;GET ITERATION COUNT
 4408  3D                4.0            DCR     A               ;DOWNCOUNT
 4409  32  9A  4F       13.0            STA     ITERA           ;SAVE COUNT
 440C  F2  A2  43       10.0            JP      TST03X          ;DO TEST UNTIL TILL = 0
                                                ;
 440F  C3  55  44       10.0            JMP     TEST4           ;GO DO THE NEXT TEST
                                                ;
 4412  3E  00            7.0    STATCK: MVI     A,0             ;LOAD THE READ PATH ENABLE COMMAND
 4414  D3  09           10.0            OUT     RPCTL           ;ENABLE THE READ PATH
                                                ;
 4416  3E  01            7.0            MVI     A,@001          ;LOAD THE EXPECTED STATUS BYTE
 4418  32  42  4B       13.0            STA     RUNSTA          ;SAVE IN MEMORY
                                                ;
 441B  3E  0A            7.0            MVI     A,RRCMT         ;LOAD THE BASE COMMAND BYTE
 441D  32  43  4B       13.0            STA     RUNCMD          ;SAVE IN MEMORY
                                                ;
 4420  3A  43  4B       13.0    STALP1: LDA     RUNCMD          ;GET THE COMMAND BYTE
 4423  D3  0B           10.0            OUT     RCMD            ;ISSUE THE COMMAND
                                                ;
```

```
1571  4425  00                  4.0           NOP                         ;WAIT
1572  4426  00                  4.0           NOP                         ;WAIT
1573  4427  00                  4.0           NOP                         ;WAIT
1574  4428  00                  4.0           NOP                         ;WAIT
1575                                                                      ;
1576  4429  DB  02              10.0           IN      RSTAT               ;GET THE STATUS BYTE
1577  442B  21  42  4B          10.0           LXI     H,RUNSTA            ;LOAD RUN TIME COMMAND ADDRESS
1578  442E  BE                   7.0           CMP     M                   ;EQUAL?
1579  442F  CA  3F  44           10.0           JZ      STACN1              ;YES-CONTINUE
1580  4432                                      ROUT    ADATA               ;NO-SAVE ACTUAL DATA IN CAS
 (1)  4432  D3  94               10.0                   OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  4434  7F                    4.0                   MOV     A,A                 ;RETRY LINK
1581  4435  3A  42  4B          13.0           LDA     RUNSTA              ;GET EXPECTED STATUS
1582  4438                                      ROUT    EDATA               ;SAVE EXPECTED DATA IN CAS
 (1)  4438  D3  95               10.0                   OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  443A  7F                    4.0                   MOV     A,A                 ;RETRY LINK
1583  443B  37                    4.0           STC                         ;SET CARRY FOR ERROR FLAG
1584  443C  C3  54  44           10.0           JMP     RETURN              ;GO REPORT
1585
1586  443F  3A  42  4B          13.0  STACN1:  LDA     RUNSTA              ;GET THE RUNTIME STATUS
1587  4442  A7                    4.0           ANA     A                   ;CLEAR CARRY
1588  4443  17                    4.0           RAL                         ;SHIFT THE DATA LEFT
1589  4444  32  42  4B          13.0           STA     RUNSTA              ;SAVE UPDATED RUN TIME STATUS
1590  4447  3A  43  4B          13.0           LDA     RUNCMD              ;GET THE RUNTIME COMMAND
1591  444A  C6  10                7.0           ADI     @020                ;ADD INCREMENTAL VALUE
1592  444C  32  43  4B          13.0           STA     RUNCMD              ;SAVE THE NEW COMMAND
1593  444F  FE  8A                7.0           CPI     @212                ;DONE?
1594  4451  C2  20  44           10.0           JNZ     STALP1              ;NO-KEEP IT GOING
1595  4454  C9                   10.0  RETURN:  RET                         ;YES-END OF TEST
1596
```

```
1598                                    .SBTTL  TEST 04 - READ PATH SELF TEST
1599
1600  4455                              ST
 (1)                                    ;*****************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1601                                    ;*RPM - SELF TEST - TEST
1602  4455                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1603                                    ;*THIS TEST COMMANDS THE READ PATH MICRO TO RUN ITS SELF TEST DIAGNOSTIC.
1604  4455                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1605                                    ;*BGNTST
1606                                    ;*  SET NORMAL READ PATH CLOCK
1607                                    ;*  ENABLE THE READ PATH
1608                                    ;*  OUTPUT THE READ PATH MICRO SELF TEST COMMAND
1609                                    ;*  DELAY 10 MICRO SECONDS
1610                                    ;*  INPUT THE READ PATH STATUS
1611                                    ;*  IF INPUT STATUS = ''RPOK'' (103)
1612                                    ;*  :  THEN-CONTINUE
1613                                    ;*  :  ELSE-ERROR
1614                                    ;*  ENDIF
1615                                    ;*  IF DATA READY, STATUS VALID, PREAMBLE ERROR AND BOP=1
1616                                    ;*    AND CLOCK STOPPED AND STATISTICS/NORMAL=0
1617                                    ;*  :  THEN-CONTINUE
1618                                    ;*  :  ELSE-ERROR
1619                                    ;*  ENDIF
1620                                    ;*  CALL SUBROUTINE RCCLR
1621                                    ;*  SET STOP CLOCK AND STATISTICS/NORMAL
1622                                    ;*  IF CLOCK STOPPED AND STATISTICS/NORMAL=1
1623                                    ;*  :  THEN-CONTINUE
1624                                    ;*  :  ELSE-ERROR
1625                                    ;*  ENDIF
1626                                    ;*  ENABLE THE READ PATH CLOCK
1627                                    ;*  OUTPUT THE READ CHANNEL SELF TEST COMMAND
1628                                    ;*  DELAY 75 MICRO SECONDS
1629                                    ;*  INPUT THE READ PATH STATUS
1630                                    ;*  IF READ PATH STATUS=''RCHOK'' (106)
1631                                    ;*  :  THEN-CONTINUE
1632                                    ;*  :  ELSE-ERROR
1633                                    ;*  ENDIF
1634                                    ;*ENDTST
1635  4455                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1636                                    ;*RPM1 MICRO TEST 04
1637                                    ;*RPM1 MICRO ERROR 04
1638                                    ;*RPM1-READ PATH SELF TEST
1639                                    ;*M8953, M8960
```

```
                                        ;*SELF DIAGNOSIS FAILURE
                                        ;*
                                        ;*RPM1 MICRO TEST 04
                                        ;*RPM1 MICRO ERROR 05
                                        ;*RPM1-READ PATH-SELF TEST
                                        ;*M8953, M8960
                                        ;*READ PATH IF. REGISTER 01 INCORRECT AFTER SELF TEST
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 04
                                        ;*RPM1 MICRO ERROR 06
                                        ;*RPM1-READ PATH-SELF TEST
                                        ;*M8953, M8960
                                        ;*READ PATH IF. REGISTER 01 INCORRECT AFTER CLEAR
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 04
                                        ;*RPM1 MICRO ERROR 07
                                        ;*RPM1-READ PATH-SELF TEST
                                        ;*M8953, M8960
                                        ;*READ CHANNEL INTERFACE FAILURE
4455                                    S
 (1)                                    ; ***************************************************************************
                                        
4455                                    TEST4:  TESTX   @4              ;INITIALIZE THE TEST
 (1) 4455  3E  04            7.0                MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1) 4457  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-READ PATH-SELF TEST
                                        ;&M8953, M8960
                                        
445A  3E  10                7.0  TST04X: MVI     A,RDCLK
445C  D3  F0               10.0         OUT     CLKCTL          ;SET NORMAL CLOCKS
                                        
445E  3E  00                7.0         MVI     A,0             ;LOAD THE READ PATH ENABLE BIT
4460  D3  09               10.0         OUT     RPCTL           ;ENABLE THE READ PATH
4462  3E  08                7.0  RMCLP1: MVI     A,RMCTST        ;LOAD THE RMC SELF TEST COMMAND
4464  D3  0B               10.0         OUT     RCMD            ;ISSUE THE COMMAND
                                                                 ;
4466  3E  02                7.0         MVI     A,@2            ;WAIT 10 MICRO SECONDS
4468  3D                    4.0  TLOOP0: DCR     A
4469  C2  68  44           10.0         JNZ     TLOOP0
                                                                 ;
446C  DB  02               10.0         IN      RSTAT           ;GET THE SELF TEST STATUS
446E  FE  43                7.0         CPI     RPOK            ;PASS THE SELF TEST?
4470  CA  78  44           10.0         JZ      RMCCN1          ;YES-CONTINUE
```

```
1685  4473                                          ERR     RMCLP1,RMCCN1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4473  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4476  07                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4477  00                                              .BYTE
 (1)  4478  CD  15  28        18.0                  RMCCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  447B  DA  62  44        10.0                          JC      RMCLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1686                                        ;>SELF DIAGNOSIS FAILURE
1687  447E  3E  18                7.0       MVI     A,$18           ;SAVE THE EXPECTED DATA
1688  4480                                    ROUT    EDATA
 (1)  4480  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4482  7F                     4.0                  MOV     A,A             ;RETRY LINK
1689  4483  DB  01                10.0      IN      RPCHI           ;GET READ PATH REGISTER
1690  4485  E6  7E                 7.0       ANI     $7E             ;SAVE WANTED BITS
1691  4487                                    ROUT    ADATA
 (1)  4487  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4489  7F                     4.0                  MOV     A,A             ;RETRY LINK
1692  448A  FE  78                 7.0       CPI     $78             ;DATA READY, STAT VALID,PREAM ERR AND BOP 1
1693  448C  CA  94  44            10.0       JZ      RMCCN2          ;CLOCK STOPPED AND STATISTICS/NORMAL=0
1694  448F                                    ERRB    RMCLP1,RMCCN2
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  448F  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4492  08                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4493  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4494  CD  15  28        18.0                  RMCCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4497  DA  62  44        10.0                          JC      RMCLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1695                                        ;>READ PATH IF. REGISTER 01 INCORRECT AFTER SELF TEST
1696  449A  CD  19  4A            18.0      CALL    RCCLR           ;ISSUE CLEAR ALL COMMAND
1697  449D  3E  03                 7.0       MVI     A,$03           ;SET STOP CLOCK AND STATISTICS/NORMAL
1698  449F  D3  09                10.0       OUT     RPCTL
1699  44A1  3E  06                 7.0       MVI     A,$06
1700  44A3                                    ROUT    EDATA
 (1)  44A3  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44A5  7F                     4.0                  MOV     A,A             ;RETRY LINK
1701  44A6  DB  01                10.0      IN      RPCHI
1702  44A8  E6  66                 7.0       ANI     $66
1703  44AA  FE  06                 7.0       CPI     $06
1704  44AC  CA  B7  44            10.0       JZ      RMCCN3
1705  44AF                                    ROUT    ADATA
 (1)  44AF  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44B1  7F                     4.0                  MOV     A,A             ;RETRY LINK
```

```
1707  44B2                                                  ERRB    RMCLP1,RMCCN3
 (1)                                                        ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  44B2  CD  12  28        18.0                                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B5  09                                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B6  00                                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44B7  CD  15  28        18.0                          RMCCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44BA  DA  62  44        10.0                                  JC      RMCLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1708                                                    ;>READ PATH IF. REGISTER 01 INCORRECT AFTER CLEAR
1709  44BD  3E  00               7.0                RLOOP1: MVI     A,0
1710  44BF  D3  09              10.0                        OUT     RPCTL           ;RE-ENABLE THE READ PATH CLOCK
1711  44C1  3E  0C               7.0                        MVI     A,RCHTST        ;ISSUE THE READ CHANNEL
1712  44C3  D3  0B              10.0                        OUT     RCMD            ;SELF TEST COMMAND
1713                                                    ;
1714                                                    ;PERFORM A 75 MICROSECOND TIMEOUT
1715                                                    ;
1716  44C5  3E  0F               7.0                        MVI     A,15            ;DELAY 75 MICROSECONDS
1717  44C7  3D                   4.0                TLOOP4: DCR     A
1718  44C8  C2  C7  44          10.0                        JNZ     TLOOP4
1719  44CB  DB  02              10.0                        IN      RSTAT           ;INPUT THE READ PATH STATUS
1720  44CD  FE  46               7.0                        CPI     RCHOK
1721  44CF  CA  D7  44          10.0                        JZ      RMCCN4
1722  44D2                                                  ERR     RLOOP1,RMCCN4
 (1)                                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44D2  CD  09  28        18.0                                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D5  0A                                                      .BYTE   MSGN            ,MESSAGE NUMBER ID
 (1)  44D6  00                                                      .BYTE
 (1)  44D7  CD  15  28        18.0                          RMCCN4::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44DA  DA  BD  44        10.0                                  JC      RLOOP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1723                                                    ;>READ CHANNEL INTERFACE FAILURE
1724  44DD                                                  ENDTST  TST04X
 (1)                                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44DD                                                  REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44DD  CD  06  28        18.0                                                  CALL    REQST
 (2)  44E0  00                                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  44E1  00  00                                                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  44E3  00  00                                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44E5  00                                                                      .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  44E6  07                                                                      .BYTE   7               ;REQUEST CODE
 (1)  44E7  3A  9A  4F        13.0                                  LDA     ITERA           ;GET ITERATION COUNT
 (1)  44EA  3D                 4.0                                  DCR     A               ;DOWNCOUNT
 (1)  44EB  32  9A  4F        13.0                                  STA     ITERA           ;SAVE COUNT
 (1)  44EE  F2  5A  44        10.0                                  JP      TST04X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 05 - READ CHANNEL 0 SELF TEST
44F1                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-CHANNEL 0 SELF TEST
44F1                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (001)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (001)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
44F1                            SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 27
                                ;*PPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 0 SELF TEST
```

```
1771                                    ;*M8950
1772                                    ;*MARK TWO BIT INCORRECT
1773                                    ;*EXPECTED = NNNN
1774                                    ;*
1775                                    ;*RPM1 MICRO TEST 05
1776                                    ;*RPM1 MICRO ERROR 31
1777                                    ;*RPM1-CHANNEL 0 SELF TEST
1778                                    ;*M8950
1779                                    ;*DONE NOT BIT INCORRECT
1780                                    ;*EXPECTED = NNNN
1781  44F1                              $
 (1)                                    ; ************************************************************************************
1782
1783  44F1                              TEST05: TESTX   @05             ;INITIALIZE THE TEST
 (1)  44F1  3E  05          7.0                     MVI     A,@05       ;DEFINE THE TEST NUMBER
 (1)  44F3  CD  03  28     18.0                     CALL    TSET        ;SETUP THE TEST
1784                                    ;%RPM1-CHANNEL 0 SELF TEST
1785                                    ;&M8950, M8953
1786
1787  44F6  3E  01          7.0  TST05X: MVI     A,@001          ;LOAD THE CHANNEL MASK
1788  44F8  32  4B  4B       13.0         STA     RCMSK
1789  44FB  CD  19  4A       18.0         CALL    RCCLR           ;CALL SUBROUTINE RCCLR
1790  44FE  3E  00           7.0         MVI     A,0
1791  4500  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
1792  4503  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
1793  4506  DA  F6  44       10.0         JC      TST05X
1794  4509  CD  32  4A       18.0  TST05Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
1795  450C  3E  01           7.0         MVI     A,@001
1796  450E  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
1797  4511  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
1798  4514  DA  09  45       10.0         JC      TST05Y
1799  4517                               ENDTST  TST05X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4517                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4517  CD  06  28     18.0                             CALL    REQST
 (2)  451A  00                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  451B  00  00                                          .WORD                   ;SYSTEM '=' COUNT
 (2)  451D  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  451F  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4520  07                                              .BYTE   7               ;REQUEST CODE
 (1)  4521  3A  9A  4F     13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  4524  3D              4.0         DCR     A               ;DOWNCOUNT
 (1)  4525  32  9A  4F     13.0         STA     ITERA           ;SAVE COUNT
 (1)  4528  F2  F6  44     10.0         JP      TST05X          ;DO TEST UNTIL TILL = 0
```

```
1801                                    .SBTTL  TEST 06 - READ CHANNEL 1 SELF TEST
1802  452B                              ST
 (1)                                    ;*****************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1803                                    ;*RPM1-CHANNEL 1 SELF TEST
1804  452B                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1805                                    ;*BGNTST
1806                                    ;*  LOAD THE CHANNEL MASK (002)
1807                                    ;*  CALL SUBROUTINE RCCLR
1808                                    ;*  SET THE EXPECTED DATA STATE (0)
1809                                    ;*  CALL SUBROUTINE RCCK
1810                                    ;*  CALL SUBROUTINE RCSTRT
1811                                    ;*  SET THE EXPECTED DATA STATE (002)
1812                                    ;*  CALL SUBROUTINE RCCK
1813                                    ;*ENDTST
1814  452B                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1815                                    ;*RPM1 MICRO TEST 06
1816                                    ;*RPM1 MICRO ERROR 24
1817                                    ;*RPM1-CHANNEL 1 SELF TEST
1818                                    ;*M8950
1819                                    ;*END MARK BIT INCORRECT
1820                                    ;*EXPECTED = NNNN
1821                                    ;*
1822                                    ;*RPM1 MICRO TEST 06
1823                                    ;*RPM1 MICRO ERROR 25
1824                                    ;*RPM1-CHANNEL 1 SELF TEST
1825                                    ;*M8950
1826                                    ;*DATA BIT INCORRECT
1827                                    ;*EXPECTED = NNNN
1828                                    ;*
1829                                    ;*RPM1 MICRO TEST 06
1830                                    ;*RPM1 MICRO ERROR 26
1831                                    ;*RPM1-CHANNEL 1 SELF TEST
1832                                    ;*M8950
1833                                    ;*POSTAMBLE BIT INCORRECT
1834                                    ;*EXPECTED = NNNN
1835                                    ;*
1836                                    ;*RPM1 MICRO TEST 06
1837                                    ;*RPM1 MICRO ERROR 27
1838                                    ;*RPM1-CHANNEL 1 SELF TEST
1839                                    ;*M8950
1840                                    ;*ILLEGAL BIT INCORRECT
1841                                    ;*EXPECTED = NNNN
1842                                    ;*
1843                                    ;*RPM1 MICRO TEST 06
1844                                    ;*RPM1 MICRO ERROR 30
1845                                    ;*RPM1-CHANNEL 1 SELF TEST
```

```
1846                                    ;*M8950
1847                                    ;*MARK TWO BIT INCORRECT
1848                                    ;*EXPECTED = NNNN
1849                                    ;*
1850                                    ;*RPM1 MICRO TEST 06
1851                                    ;*RPM1 MICRO ERROR 31
1852                                    ;*RPM1-CHANNEL 1 SELF TEST
1853                                    ;*M8950
1854                                    ;*DONE NOT BIT INCORRECT
1855                                    ;*EXPECTED = NNNN
1856  452B                              S
 (1)                                    ; ***************************************************************************
1857
1858  452B                              TEST06: TESTX   @06             ;INITIALIZE THE TEST
 (1)  452B  3E  06          7.0                     MVI     A,@06       ;DEFINE THE TEST NUMBER
 (1)  452D  CD  03  28     18.0                     CALL    TSET        ;SETUP THE TEST
1859                                    ;%RPM1-CHANNEL 1 SELF TEST
1860                                    ;&M8950, M8953
1861
1862  4530  3E  02          7.0  TST06X: MVI     A,@002          ;LOAD THE CHANNEL MASK
1863  4532  32  4B  4B       13.0         STA     RCMSK
1864  4535  CD  19  4A       18.0         CALL    RCCLR           ;CALL SUBROUTINE RCCLR
1865  4538  3E  00           7.0         MVI     A,0
1866  453A  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
1867  453D  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
1868  4540  DA  30  45       10.0         JC      TST06X
1869  4543  CD  32  4A       18.0  TST06Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
1870  4546  3E  02           7.0         MVI     A,@002
1871  4548  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
1872  454B  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
1873  454E  DA  43  45       10.0         JC      TST06Y
1874  4551                              ENDTST  TST06X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4551                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4551  CD  06  28     18.0                     CALL    REQST
 (2)  4554  00                                      .BYTE           ;DATA PATTERN NUMBER
 (2)  4555  00  00                                  .WORD           ;SYSTEM '@' COUNT
 (2)  4557  00  00                                  .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4559  00                                      .BYTE           ;DATA COMPARE FLAG IF -1
 (2)  455A  07                                      .BYTE   7       ;REQUEST CODE
 (1)  455B  3A  9A  4F     13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  455E  3D              4.0         DCR     A               ;DOWNCOUNT
 (1)  455F  32  9A  4F     13.0         STA     ITERA           ;SAVE COUNT
 (1)  4562  F2  30  45     10.0         JP      TST06X          ;DO TEST UNTIL TILL = 0
```

```
1876                                    .SBTTL  TEST 07 - READ CHANNEL 2 SELF TEST
1877  4565                              ST
 (1)                                    ;*********************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1878                                    ;*RPM1-CHANNEL 2 SELF TEST
1879  4565                              SP
 (1)                                    ;*********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1880                                    ;*BGNTST
1881                                    ;*  LOAD THE CHANNEL MASK (004)
1882                                    ;*  CALL SUBROUTINE RCCLR
1883                                    ;*  SET THE EXPECTED DATA STATE (0)
1884                                    ;*  CALL SUBROUTINE RCCK
1885                                    ;*  CALL SUBROUTINE RCSTRT
1886                                    ;*  SET THE EXPECTED DATA STATE (004)
1887                                    ;*  CALL SUBROUTINE RCCK
1888                                    ;*ENDTST
1889  4565                              SE
 (1)                                    ;*********************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1890                                    ;*RPM1 MICRO TEST 07
1891                                    ;*RPM1 MICRO ERROR 24
1892                                    ;*RPM1-CHANNEL 2 SELF TEST
1893                                    ;*M8950
1894                                    ;*END MARK BIT INCORRECT
1895                                    ;*EXPECTED = NNNN
1896                                    ;*
1897                                    ;*RPM1 MICRO TEST 07
1898                                    ;*RPM1 MICRO ERROR 25
1899                                    ;*RPM1-CHANNEL 2 SELF TEST
1900                                    ;*M8950
1901                                    ;*DATA BIT INCORRECT
1902                                    ;*EXPECTED = NNNN
1903                                    ;*
1904                                    ;*RPM1 MICRO TEST 07
1905                                    ;*RPM1 MICRO ERROR 26
1906                                    ;*RPM1-CHANNEL 2 SELF TEST
1907                                    ;*M8950
1908                                    ;*POSTAMBLE BIT INCORRECT
1909                                    ;*EXPECTED = NNNN
1910                                    ;*
1911                                    ;*RPM1 MICRO TEST 07
1912                                    ;*RPM1 MICRO ERROR 27
1913                                    ;*RPM1-CHANNEL 2 SELF TEST
1914                                    ;*M8950
1915                                    ;*ILLEGAL BIT INCORRECT
1916                                    ;*EXPECTED = NNNN
1917                                    ;*
1918                                    ;*RPM1 MICRO TEST 07
1919                                    ;*RPM1 MICRO ERROR 30
1920                                    ;*RPM1-CHANNEL 2 SELF TEST
```

```
 1921                                   ;*M8950
 1922                                   ;*MARK TWO BIT INCORRECT
 1923                                   ;*EXPECTED = NNNN
 1924                                   ;*
 1925                                   ;*RPM1 MICRO TEST 07
 1926                                   ;*RPM1 MICRO ERROR 31
 1927                                   ;*RPM1-CHANNEL 2 SELF TEST
 1928                                   ;*M8950
 1929                                   ;*DONE NOT BIT INCORRECT
 1930                                   ;*EXPECTED = NNNN
 1931  4565                             S
  (1)                                   ; ***************************************************************************
 1932
 1933  4565                             TEST07: TESTX   @07              ;INITIALIZE THE TEST
  (1)  4565  3E  07          7.0                  MVI     A,@07          ;DEFINE THE TEST NUMBER
  (1)  4567  CD  03  28     18.0                  CALL    TSET           ;SETUP THE TEST
 1934                                   ;%RPM1-CHANNEL 2 SELF TEST
 1935                                   ;&M8950, M8953
 1936
 1937  456A  3E  04          7.0  TST07X: MVI     A,@004         ;LOAD THE CHANNEL MASK
 1938  456C  32  4B  4B       13.0         STA     RCMSK
 1939  456F  CD  19  4A       18.0         CALL    RCCLR          ;CALL SUBROUTINE RCCLR
 1940  4572  3E  00           7.0         MVI     A,0
 1941  4574  32  4A  4B       13.0         STA     RCEDAT         ;SET THE EXPECTED DATA STATE
 1942  4577  CD  4B  4A       18.0         CALL    RCCK           ;CALL SUBROUTINE RCCK
 1943  457A  DA  6A  45       10.0         JC      TST07X
 1944  457D  CD  32  4A       18.0  TST07Y: CALL    RCSTRT         ;CALL SUBROUTINE RCSTRT
 1945  4580  3E  04           7.0         MVI     A,@004
 1946  4582  32  4A  4B       13.0         STA     RCEDAT         ;SET THE EXPECTED DATA STATE
 1947  4585  CD  4B  4A       18.0         CALL    RCCK           ;CALL SUBROUTINE RCCK
 1948  4588  DA  7D  45       10.0         JC      TST07Y
 1949  4588                             ENDTST  TST07X
  (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  458B                          REQ     7              ;FAKE CALL TO KEEP TEST ALIVE
  (2)  458B  CD  06  28     18.0                  CALL    REQST
  (2)  458E  00                                   .BYTE                  ;DATA PATTERN NUMBER
  (2)  458F  00  00                               .WORD                  ;SYSTEM '*' COUNT
  (2)  4591  00  00                               .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4593  00                                   .BYTE                  ;DATA COMPARE FLAG IF =1
  (2)  4594  07                                   .BYTE   7              ;REQUEST CODE
  (1)  4595  3A  9A  4F     13.0         LDA     ITERA          ;GET ITERATION COUNT
  (1)  4598  3D              4.0         DCR     A              ;DOWNCOUNT
  (1)  4599  32  9A  4F     13.0         STA     ITERA          ;SAVE COUNT
  (1)  459C  F2  6A  45     10.0         JP      TST07X         ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - READ CHANNEL 3 SELF TEST
459F                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-CHANNEL 3 SELF TEST
459F                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (010)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (010)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
459F                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 10
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 3 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 10
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 3 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 10
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 3 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 10
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 3 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 10
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 3 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 10
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 3 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
459F                                    S
 (1)                                    ; ************************************************************************************
                                        TEST10: TESTX   @10             ;INITIALIZE THE TEST
459F  3E  08            7.0                     MVI     A,@10           ;DEFINE THE TEST NUMBER
45A1  CD  03  28        18.0                    CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 3 SELF TEST
                                        ;&M8950, M8953

45A4  3E  08            7.0     TST10X: MVI     A,@010          ;LOAD THE CHANNEL MASK
45A6  32  4B  4B        13.0            STA     RCMSK
45A9  CD  19  4A        18.0            CALL    RCCLR           ;CALL SUBROUTINE RCCLR
45AC  3E  00            7.0             MVI     A,0
45AE  32  4A  4B        13.0            STA     RCEDAT          ;SET THE EXPECTED DATA STATE
45B1  CD  4B  4A        18.0            CALL    RCCK            ;CALL SUBROUTINE RCCK
45B4  DA  A4  45        10.0            JC      TST10X
45B7  CD  32  4A        18.0    TST10Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
45BA  3E  08            7.0             MVI     A,@010
45BC  32  4A  4B        13.0            STA     RCEDAT          ;SET THE EXPECTED DATA STATE
45BF  CD  4B  4A        18.0            CALL    RCCK            ;CALL SUBROUTINE RCCK
45C2  DA  B7  45        10.0            JC      TST10Y
45C5                                    ENDTST  TST10X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
45C5                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
45C5  CD  06  28        18.0                            CALL    REQST
45C8  00                                                .BYTE                   ;DATA PATTERN NUMBER
45C9  00  00                                            .WORD                   ;SYSTEM "#" COUNT
45CB  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
45CD  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
45CE  07                                                .BYTE   7               ;REQUEST CODE
45CF  3A  9A  4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
45D2  3D                4.0             DCR     A               ;DOWNCOUNT
45D3  32  9A  4F        13.0            STA     ITERA           ;SAVE COUNT
45D6  F2  A4  45        10.0            JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
2026                                    .SBTTL  TEST 11 - READ CHANNEL 4 SELF TEST
2027  45D9                              ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
2028                                    ;*RPM1-CHANNEL 4 SELF TEST
2029  45D9                              SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2030                                    ;*BGNTST
2031                                    ;*  LOAD THE CHANNEL MASK (020)
2032                                    ;*  CALL SUBROUTINE RCCLR
2033                                    ;*  SET THE EXPECTED DATA STATE (0)
2034                                    ;*  CALL SUBROUTINE RCCK
2035                                    ;*  CALL SUBROUTINE RCSTRT
2036                                    ;*  SET THE EXPECTED DATA STATE (020)
2037                                    ;*  CALL SUBROUTINE RCCK
2038                                    ;*ENDTST
2039  45D9                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2040                                    ;*RPM1 MICRO TEST 11
2041                                    ;*RPM1 MICRO ERROR 24
2042                                    ;*RPM1-CHANNEL 4 SELF TEST
2043                                    ;*M8950
2044                                    ;*END MARK BIT INCORRECT
2045                                    ;*EXPECTED = NNNN
2046                                    ;*
2047                                    ;*RPM1 MICRO TEST 11
2048                                    ;*RPM1 MICRO ERROR 25
2049                                    ;*RPM1-CHANNEL 4 SELF TEST
2050                                    ;*M8950
2051                                    ;*DATA BIT INCORRECT
2052                                    ;*EXPECTED = NNNN
2053                                    ;*
2054                                    ;*RPM1 MICRO TEST 11
2055                                    ;*RPM1 MICRO ERROR 26
2056                                    ;*RPM1-CHANNEL 4 SELF TEST
2057                                    ;*M8950
2058                                    ;*POSTAMBLE BIT INCORRECT
2059                                    ;*EXPECTED = NNNN
2060                                    ;*
2061                                    ;*RPM1 MICRO TEST 11
2062                                    ;*RPM1 MICRO ERROR 27
2063                                    ;*RPM1-CHANNEL 4 SELF TEST
2064                                    ;*M8950
2065                                    ;*ILLEGAL BIT INCORRECT
2066                                    ;*EXPECTED = NNNN
2067                                    ;*
2068                                    ;*RPM1 MICRO TEST 11
2069                                    ;*RPM1 MICRO ERROR 30
2070                                    ;*RPM1-CHANNEL 4 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 11
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 4 SELF TEST
                                        ;*M8950
                                        ;*DONE ?? BIT INCORRECT
                                        ;*EXPECTED = NNNN
2081  45D9                              $
 (1)                                    ; ***************************************************************************************
2082
2083  45D9                              TEST11: TESTX   @11                   ;INITIALIZE THE TEST
 (1)  45D9  3E  09            7.0                        MVI     A,@11         ;DEFINE THE TEST NUMBER
 (1)  45DB  CD  03  28       18.0                        CALL    TSET          ;SETUP THE TEST
2084                                    ;%RPM1-CHANNEL 4 SELF TEST
2085                                    ;&M8950, M8953
2086
2087  45DE  3E  10                7.0   TST11X: MVI     A,@020                ;LOAD THE CHANNEL MASK
2088  45E0  32  4B  4B           13.0           STA     RCMSK
2089  45E3  CD  19  4A           18.0           CALL    RCCLR                 ;CALL SUBROUTINE RCCLR
2090  45E6  3E  00                7.0           MVI     A,0
2091  45E8  32  4A  4B           13.0           STA     RCEDAT                ;SET THE EXPECTED DATA STATE
2092  45EB  CD  4B  4A           18.0           CALL    RCCK                  ;CALL SUBROUTINE RCCK
2093  45EE  DA  DE  45           10.0           JC      TST11X
2094  45F1  CD  32  4A           18.0   TST11Y: CALL    RCSTRT                ;CALL SUBROUTINE RCSTRT
2095  45F4  3E  10                7.0           MVI     A,@020
2096  45F6  32  4A  4B           13.0           STA     RCEDAT                ;SET THE EXPECTED DATA STATE
2097  45F9  CD  4B  4A           18.0           CALL    RCCK                  ;CALL SUBROUTINE RCCK
2098  45FC  DA  F1  45           10.0           JC      TST11Y
2099  45FF                                  ENDTST  TST11X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45FF                                  REQ     7                     ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45FF  CD  06  28       18.0                        CALL    REQST
 (2)  4602  00                                           .BYTE               ;DATA PATTERN NUMBER
 (2)  4603  00  00                                       .WORD               ;SYSTEM '*' COUNT
 (2)  4605  00  00                                       .WORD               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4607  00                                           .BYTE               ;DATA COMPARE FLAG IF -1
 (2)  4608  07                                           .BYTE   7           ;REQUEST CODE
 (1)  4609  3A  9A  4F       13.0           LDA     ITERA                 ;GET ITERATION COUNT
 (1)  460C  3D                4.0           DCR     A                     ;DOWNCOUNT
 (1)  460D  32  9A  4F       13.0           STA     ITERA                 ;SAVE COUNT
 (1)  4610  F2  DE  45       10.0           JP      TST11X                ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - READ CHANNEL 5 SELF TEST
4613                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-CHANNEL 5 SELF TEST
4613                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (040)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (040)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
4613                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 5 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 12
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 5 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
4613                                    S
(1)                                     ; ************************************************************************************
                                        TEST12: TESTX   @12             ;INITIALIZE THE TEST
(1) 4613 3E 0A          7.0                             MVI     A,@12           ;DEFINE THE TEST NUMBER
(1) 4615 CD 03 28       18.0                            CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 5 SELF TEST
                                        ;&M8950, M8953
4618 3E 20              7.0     TST12X: MVI     A,@040          ;LOAD THE CHANNEL MASK
461A 32 4B 4B           13.0            STA     RCMSK
461D CD 19 4A           18.0            CALL    RCCLR           ;CALL SUBROUTINE RCCLR
4620 3E 00              7.0             MVI     A,0
4622 32 4A 4B           13.0            STA     RCEDAT          ;SET THE EXPECTED DATA STATE
4625 CD 4B 4A           18.0            CALL    RCCK            ;CALL SUBROUTINE RCCK
4628 DA 18 46           10.0            JC      TST12X
462B CD 32 4A           18.0    TST12Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
462E 3E 20              7.0             MVI     A,@040
4630 32 4A 4B           13.0            STA     RCEDAT          ;SET THE EXPECTED DATA STATE
4633 CD 4B 4A           18.0            CALL    RCCK            ;CALL SUBROUTINE RCCK
4636 DA 2B 46           10.0            JC      TST12Y
4639                                    ENDTST  TST12X
(1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
(2) 4639                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
(2) 4639 CD 06 28       18.0                                    CALL    REQST
(2) 463C 00                                                     .BYTE                   ;DATA PATTERN NUMBER
(2) 463D 00 00                                                  .WORD                   ;SYSTEM '=' COUNT
(2) 463F 00 00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2) 4641 00                                                     .BYTE                   ;DATA COMPARE FLAG IF -1
(2) 4642 07                                                     .BYTE   7               ;REQUEST CODE
(1) 4643 3A 9A 4F       13.0            LDA     ITERA           ;GET ITERATION COUNT
(1) 4646 3D             4.0             DCR     A               ;DOWNCOUNT
(1) 4647 32 9A 4F       13.0            STA     ITERA           ;SAVE COUNT
(1) 464A F2 18 46       10.0            JP      TST12X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - READ CHANNEL 6 SELF TEST
                               ST
                               ; ************************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*RPM1-CHANNEL 6 SELF TEST
                               SP
                               ; ************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*  LOAD THE CHANNEL MASK (100)
                               ;*  CALL SUBROUTINE RCCLR
                               ;*  SET THE EXPECTED DATA STATE (0)
                               ;*  CALL SUBROUTINE RCCK
                               ;*  CALL SUBROUTINE RCSTRT
                               ;*  SET THE EXPECTED DATA STATE (100)
                               ;*  CALL SUBROUTINE RCCK
                               ;*ENDTST
                               SE
                               ; ************************************************************************
                               ;*ERRORS
                               ;*------
                               ;*RPM1 MICRO TEST 13
                               ;*RPM1 MICRO ERROR 24
                               ;*RPM1-CHANNEL 6 SELF TEST
                               ;*M8950
                               ;*END MARK BIT INCORRECT
                               ;*EXPECTED = NNNN
                               ;*
                               ;*RPM1 MICRO TEST 13
                               ;*RPM1 MICRO ERROR 25
                               ;*RPM1-CHANNEL 6 SELF TEST
                               ;*M8950
                               ;*DATA BIT INCORRECT
                               ;*EXPECTED = NNNN
                               ;*
                               ;*RPM1 MICRO TEST 13
                               ;*RPM1 MICRO ERROR 26
                               ;*RPM1-CHANNEL 6 SELF TEST
                               ;*M8950
                               ;*POSTAMBLE BIT INCORRECT
                               ;*EXPECTED = NNNN
                               ;*
                               ;*RPM1 MICRO TEST 13
                               ;*RPM1 MICRO ERROR 27
                               ;*RPM1-CHANNEL 6 SELF TEST
                               ;*M8950
                               ;*ILLEGAL BIT INCORRECT
                               ;*EXPECTED = NNNN
                               ;*
                               ;*RPM1 MICRO TEST 13
                               ;*RPM1 MICRO ERROR 30
                               ;*RPM1-CHANNEL 6 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 13
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 6 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        S
                                        ; ************************************************************************************

                                        TEST13: TESTX   @13             ;INITIALIZE THF TEST
                                                        MVI     A,@13           ;DEFINE THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 6 SELF TEST
                                        ;&M8950, M8953

                                        TST13X: MVI     A,@100          ;LOAD THE CHANNEL MASK
                                                STA     RCMSK
                                                CALL    RCCLR           ;CALL SUBROUTINE RCCLR
                                                MVI     A,0
                                                STA     RCEDAT          ;SET THE EXPECTED DATA STATE
                                                CALL    RCCK            ;CALL SUBROUTINE RCCK
                                                JC      TST13X
                                        TST13Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
                                                MVI     A,@100
                                                STA     RCEDAT          ;SET THE EXPECTED DATA STATE
                                                CALL    RCCK            ;CALL SUBROUTINE RCCK
                                                JC      TST13Y
                                                ENDTST  TST13X
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
                                                                CALL    REQST
                                                                .BYTE                   ;DATA PATTERN NUMBER
                                                                .WORD                   ;SYSTEM '=' COUNT
                                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                                .BYTE                   ;DATA COMPARE FLAG IF  1
                                                                .BYTE   7               ;REQUEST CODE
                                                LDA     ITERA           ;GET ITERATION COUNT
                                                DCR     A               ;DOWNCOUNT
                                                STA     ITERA           ;SAVE COUNT
                                                JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - READ CHANNEL 7 SELF TEST
2252  4687                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-CHANNEL 7 SELF TEST
2254  4687                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (200)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (200)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
2264  4687                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 7 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 14
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 7 SELF TEST
                                        ;*M3950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
4687                                    S
 (1)                                    ; ************************************************************************************

                                        TEST14: TESTX   @14             ;INITIALIZE THE TEST
 (1) 4687  3E  0C            7.0                 MVI     A,@14           ;DEFINE THE TEST NUMBER
 (1) 4689  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 7 SELF TEST
                                        ;&M8950, M8953

468C  3E  80                7.0  TST14X: MVI     A,@200          ;LOAD THE CHANNEL MASK
468E  32  4B  4B           13.0          STA     RCMSK
4691  CD  19  4A           18.0          CALL    RCCLR           ;CALL SUBROUTINE RCCLR
4694  3E  00                7.0          MVI     A,0
4696  32  4A  4B           13.0          STA     RCEDAT          ;SET THE EXPECTED DATA STATE
4699  CD  4B  4A           18.0          CALL    RCCK            ;CALL SUBROUTINE RCCK
469C  DA  8C  46           10.0          JC      TST14X
469F  CD  32  4A           18.0  TST14Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
46A2  3E  80                7.0          MVI     A,@200
46A4  32  4A  4B           13.0          STA     RCEDAT          ;SET THE EXPECTED DATA STATE
46A7  CD  4B  4A           18.0          CALL    RCCK            ;CALL SUBROUTINE RCCK
46AA  DA  9F  46           10.0          JC      TST14Y
46AD                                    ENDTST  TST14X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 46AD                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 46AD  CD  06  28       18.0                         CALL    REQST
 (2) 46B0  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2) 46B1  00  00                                        .WORD                   ;SYSTEM '*' COUNT
 (2) 46B3  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 46B5  00                                            .BYTE                   ;DATA COMPARE FLAG IF -1
 (2) 46B6  07                                            .BYTE   7               ;REQUEST CODE
 (1) 46B7  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1) 46BA  3D                4.0          DCR     A               ;DOWNCOUNT
 (1) 46BB  32  9A  4F       13.0          STA     ITERA           ;SAVE COUNT
 (1) 46BE  F2  8C  46       10.0          JP      TST14X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 15 - READ CHANNEL P SELF TEST
2327  46C1                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM1-CHANNEL P SELF TEST
2329  46C1                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (001)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCKP
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (001)
                                ;*  CALL SUBROUTINE RCCKP
                                ;*ENDTST
2339  46C1                      SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 15
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL P SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 15
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL P SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 15
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL P SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 15
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL P SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 15
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL P SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 15
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL P SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        $
                                        ; ***************************************************************************
                                        TEST15: TESTX   @15             ;INITIALIZE THE TEST
                                                        MVI     A,@15           ;DEFINE THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL P SELF TEST
                                        ;&M8950, M8953

                                        TST15X: MVI     A,@001          ;LOAD THE CHANNEL MASK
                                                STA     RCMSK
                                                CALL    RCCLR           ;CALL SUBROUTINE RCCLR
                                                MVI     A,0
                                                STA     RCEDAT          ;SET THE EXPECTED DATA STATE
                                                CALL    RCCKP           ;CALL SUBROUTINE RCCK
                                                JC      TST15X
                                        TST15Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
                                                MVI     A,@001
                                                STA     RCEDAT          ;SET THE EXPECTED DATA STATE
                                                CALL    RCCKP           ;CALL SUBROUTINE RCCK
                                                JC      TST15Y
                                                ENDTST  TST15X
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
                                                                CALL    REQST
                                                                .BYTE           ;DATA PATTERN NUMBER
                                                                .WORD           ;SYSTEM '*' COUNT
                                                                .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                                .BYTE           ;DATA COMPARE FLAG IF  1
                                                                .BYTE   7       ;REQUEST CODE
                                                LDA     ITERA           ;GET ITERATION COUNT
                                                DCR     A               ;DOWNCOUNT
                                                STA     ITERA           ;SAVE COUNT
                                                JP      TST15X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 16 - READ CHANNEL AMTIE TEST - FROM TU PORT #0
2402  46FB                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2404  46FB                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 0
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
2408  46FB                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 0 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #0 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
2418  46FB                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 16
                                ;*RPM1 MICRO ERROR 10
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*RPM1 MICRO TEST 16
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 16
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 16
```

```
                        ;*RPM1 MICRO ERROR 15
                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                        ;*M8950, M8953, M8955'S
                        ;*RPM1-CHANNEL 2 FAILED
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*RPM1 MICRO TEST 16
                        ;*RPM1 MICRO ERROR 16
                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                        ;*M8950, M8953, M8955'S
                        ;*RPM1-CHANNEL 3 FAILED
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*RPM1 MICRO TEST 16
                        ;*RPM1 MICRO ERROR 17
                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                        ;*M8950, M8953, M8955'S
                        ;*RPM1-CHANNEL 4 FAILED
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*RPM1 MICRO TEST 16
                        ;*RPM1 MICRO ERROR 20
                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                        ;*M8950, M8953, M8955'S
                        ;*RPM1-CHANNEL 5 FAILED
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*RPM1 MICRO TEST 16
                        ;*RPM1 MICRO ERROR 21
                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                        ;*M8950, M8953, M8955'S
                        ;*RPM1-CHANNEL 6 FAILED
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*RPM1 MICRO TEST 16
                        ;*RPM1 MICRO ERROR 22
                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                        ;*M8950, M8953, M8955'S
                        ;*RPM1-CHANNEL 7 FAILED
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*RPM1 MICRO TEST 16
                        ;*RPM1 MICRO ERROR 23
                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                        ;*M8950, M8953, M8955'S
                        ;*RPM1-CHANNEL P FAILED
                        ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
                        ;*ACTUAL = NNNN
```

```
2497  46FB                                  S
 (1)                                        ; ************************************************************************
2498
2499  46FB                                  TEST16: TESTX   @16              ;INITIALIZE THE TEST
 (1)  46FB  3E  0E              7.0                     MVI     A,@16         ;DEFINE THE TEST NUMBER
 (1)  46FD  CD  03  28         18.0                     CALL    TSET          ;SETUP THE TEST
2500                                        ;%RPM1- .EAD CHANNEL AMTIE TEST-FROM TU PORT #0
2501                                        ;&M8950, M8953, M8955'S
2502
2503  4700                                          REQ     @7,0,0,0,0
 (1)  4700  CD  06  28         18.0                     CALL    REQST
 (1)  4703  00                                          .BYTE   0             ;DATA PATTERN NUMBER
 (1)  4704  00  00                                      .WORD   0             ;SYSTEM '0' COUNT
 (1)  4706  00  00                                      .WORD   0             ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4708  00                                          .BYTE   0             ;DATA COMPARE FLAG IF -1
 (1)  4709  07                                          .BYTE   @7            ;REQUEST CODE
2504  470A                                          RIN     R12L            ;GET THE UNITS DESIRED
 (1)  470A  DB  94             10.0                     IN      R12L                ;READ R12L INTO AC
 (1)  470C  7F                  4.0                     MOV     A,A                 ;RETRY LINK
2505  470D  32  41  4B          13.0                 STA     UNITMP          ;STORE IN MEMORY
2506  4710  A7                   4.0                 ANA     A               ;SET THE CONDITION CODE
2507  4711  C2  22  47          10.0                 JNZ     TST16X          ;GO TEST THE SPECIFIED UNITS
2508
2509  4714                                          ERR     EXIT,DUMMY
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4714  CD  09  28         18.0                     CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0008                                       MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4717  08                                         .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  4718  00                                         .BYTE
 (1)  4719  CD  15  28         18.0             DUMMY:: CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  471C  DA  B8  48         10.0                     JC      EXIT                ;LOOP ADDRESS IF LOOP SPECIFIED
2510                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
2511                                        ;>FATAL ERROR - TEST ABORTED
2512  471F  C3  B8  48         10.0                 JMP     EXIT
2513  4722  3A  41  4B          13.0     TST16X: LDA     UNITMP          ;GET THE UNIT MAP
2514  4725  E6  01                7.0             ANI     @001            ;TEST UNIT 0?
2515  4727  CA  46  47          10.0             JZ      TEST17          ;NO-CHECK FOR TEST 17
2516  472A  CD  A0  49          18.0             CALL    CLEAR           ;CLEAR ALL THE TU PORTS
2517  472D  3E  00                7.0             MVI     A,0
2518  472F  CD  CB  49          18.0             CALL    RDCTL
2519
2520  4732                                          ENDTST  TST16X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4732                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4732  CD  06  28         18.0                     CALL    REQST
 (2)  4735  00                                          .BYTE                 ;DATA PATTERN NUMBER
 (2)  4736  00  00                                      .WORD                 ;SYSTEM '0' COUNT
 (2)  4738  00  00                                      .WORD                 ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  473A  00                                          .BYTE                 ;DATA COMPARE FLAG IF =1
 (2)  473B  07                                          .BYTE   7             ;REQUEST CODE
 (1)  473C  3A  9A  4F          13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1)  473F  3D                  4.0                 DCR     A               ;DOWNCOUNT
 (1)  4740  32  9A  4F          13.0                 STA     ITERA           ;SAVE COUNT
 (1)  4743  F2  22  47          10.0                 JP      TST16X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 17 - READ CHANNEL AMTIE TEST - FROM TU PORT #1
4746                            ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
4746                            SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 1
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
4746                            SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 1 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #1 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
4746                            SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 17
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 17
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 17
                                ;*RPM1 MICRO ERROR 15
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
```

```
2564                                    ;*RPM1 MICRO TEST 17
2565                                    ;*RPM1 MICRO ERROR 16
2566                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
2567                                    ;*M8950, M8953, M8955'S
2568                                    ;*RPM1-CHANNEL 3 FAILED
2569                                    ;*ACTUAL = NNNN
2570                                    ;*EXPECTED = NNNN
2571                                    ;*
2572                                    ;*RPM1 MICRO TEST 17
2573                                    ;*RPM1 MICRO ERROR 17
2574                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
2575                                    ;*M8950, M8953, M8955'S
2576                                    ;*RPM1-CHANNEL 4 FAILED
2577                                    ;*ACTUAL = NNNN
2578                                    ;*EXPECTED = NNNN
2579                                    ;*
2580                                    ;*RPM1 MICRO TEST 17
2581                                    ;*RPM1 MICRO ERROR 20
2582                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
2583                                    ;*M8950, M8953, M8955'S
2584                                    ;*RPM1-CHANNEL 5 FAILED
2585                                    ;*ACTUAL = NNNN
2586                                    ;*EXPECTED = NNNN
2587                                    ;*
2588                                    ;*RPM1 MICRO TEST 17
2589                                    ;*RPM1 MICRO ERROR 21
2590                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM T J PORT #1
2591                                    ;*M8950, M8953, M8955'S
2592                                    ;*RPM1-CHANNEL 6 FAILED
2593                                    ;*ACTUAL = NNNN
2594                                    ;*EXPECTED = NNNN
2595                                    ;*
2596                                    ;*RPM1 MICRO TEST 17
2597                                    ;*RPM1 MICRO ERROR 22
2598                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
2599                                    ;*M8950, M8953, M8955'S
2600                                    ;*RPM1-CHANNEL 7 FAILED
2601                                    ;*ACTUAL = NNNN
2602                                    ;*EXPECTED = NNNN
2603                                    ;*
2604                                    ;*RPM1 MICRO TEST 17
2605                                    ;*RPM1 MICRO ERROR 23
2606                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
2607                                    ;*M8950, M8953, M8955'S
2608                                    ;*RPM1-CHANNEL P FAILED
2609                                    ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
2610                                    ;*ACTUAL = NNNN
2611  4746                              S
 (1)                                    ; ***************************************************************************************
2612  4746                              TEST17: TESTX   @17             ;INITIALIZE THE TEST
 (1)  4746  3E  0F            7.0                 MVI     A,@17           ;DEFINE THE TEST NUMBER
 (1)  4748  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
2613                                    ;%RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
2614                                    ;&M8950, M8953, M8955'S
```

```
  2615  474B  3A  41  4B      13.0  TST17X: LDA     UNITMP          ;GET THE UNIT MAP
  2616  474E  E6  02              7.0          ANI     @002            ;TEST UNIT 1?
  2617  4750  CA  6F  47        10.0          JZ      TEST20          ;NO-CHECK FOR TEST 20
  2618  4753  CD  A0  49        18.0          CALL    CLEAR           ;CLEAR ALL THE TU PORTS
  2619  4756  3E  01              7.0          MVI     A,@1
  2620  4758  CD  CB  49        18.0          CALL    RDCTL
  2621  475B                                  ENDTST  TST17X
   (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
   (2)  475B                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 *  (2)  475B  CD  06  28        18.0                                  CALL    REQST
   (2)  475E  00                                                      .BYTE                   ;DATA PATTERN NUMBER
   (2)  475F  00  00                                                  .WORD                   ;SYSTEM '*' COUNT
   (2)  4761  00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
   (2)  4763  00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
   (2)  4764  07                                                      .BYTE   7               ;REQUEST CODE
   (1)  4765  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
   (1)  4768  3D                 4.0          DCR     A               ;DOWNCOUNT
   (1)  4769  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
   (1)  476C  F2  4B  47        10.0          JP      TST17X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 20 - READ CHANNEL AMTIE TEST - FROM TU PORT #2
476F                            ST
                                ;*********************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
476F                            SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 2
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
476F                            SP
                                ;*********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 2 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #2 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
476F                            SE
                                ;*********************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 20
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 20
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 20
                                ;*RPM1 MICRO ERROR 15
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
```

```
2665                                    ;*RPM1 MICRO TEST 20
2666                                    ;*RPM1 MICRO ERROR 16
2667                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2668                                    ;*M8950, M8953, M8955'S
2669                                    ;*RPM1-CHANNEL 3 FAILED
2670                                    ;*ACTUAL = NNNN
2671                                    ;*EXPECTED = NNNN
2672                                    ;*
2673                                    ;*RPM1 MICRO TEST 20
2674                                    ;*RPM1 MICRO ERROR 17
2675                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2676                                    ;*M8950, M8953, M8955'S
2677                                    ;*RPM1-CHANNEL 4 FAILED
2678                                    ;*ACTUAL = NNNN
2679                                    ;*EXPECTED = NNNN
2680                                    ;*
2681                                    ;*RPM1 MICRO TEST 20
2682                                    ;*RPM1 MICRO ERROR 20
2683                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2684                                    ;*M8950, M8953, M8955'S
2685                                    ;*RPM1-CHANNEL 5 FAILED
2686                                    ;*ACTUAL = NNNN
2687                                    ;*EXPECTED = NNNN
2688                                    ;*
2689                                    ;*RPM1 MICRO TEST 20
2690                                    ;*RPM1 MICRO ERROR 21
2691                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2692                                    ;*M8950, M8953, M8955'S
2693                                    ;*RPM1-CHANNEL 6 FAILED
2694                                    ;*ACTUAL = NNNN
2695                                    ;*EXPECTED = NNNN
2696                                    ;*
2697                                    ;*RPM1 MICRO TEST 20
2698                                    ;*RPM1 MICRO ERROR 22
2699                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2700                                    ;*M8950, M8953, M8955'S
2701                                    ;*RPM1-CHANNEL 7 FAILED
2702                                    ;*ACTUAL = NNNN
2703                                    ;*EXPECTED = NNNN
2704                                    ;*
2705                                    ;*RPM1 MICRO TEST 20
2706                                    ;*RPM1 MICRO ERROR 23
2707                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2708                                    ;*M8950, M8953, M8955'S
2709                                    ;*RPM1-CHANNEL P FAILED
2710                                    ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
2711                                    ;*ACTUAL = NNNN
2712  476F                              S
 (1)                                    ;*****************************************************************************
2713  476F                              TEST20: TESTX   @020            ;INITIALIZE THE TEST
 (1)  476F  3E  10          7.0                  MVI     A,@020          ;DEFINE THE TEST NUMBER
 (1)  4771  CD  03  28      18.0                 CALL    TSET            ;SETUP THE TEST
2714                                    ;%RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2715                                    ;&M8950, M8953, M8955'S
```

```
 2716  4774  3A  41  4B      13.0  TST020X: LDA     UNITMP          ;GET THE UNIT MAP
 2717  4777  E6  04              7.0          ANI     @004            ;TEST UNIT 2?
 2718  4779  CA  98  47        10.0          JZ      TEST21          ;NO-CHECK FOR TEST 21
 2719  477C  CD  A0  49        18.0          CALL    CLEAR           ;CLEAR ALL THE TU PORTS
 2720  477F  3E  02              7.0          MVI     A,@2
 2721  4781  CD  CB  49        18.0          CALL    RDCTL
 2722  4784                                  ENDTST  TST020X
   (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
   (2)  4784                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
   (2)  4784  CD  06  28        18.0                          CALL    REQST
   (2)  4787  00                                              .BYTE                   ;DATA PATTERN NUMBER
   (2)  4788  00  00                                          .WORD                   ;SYSTEM '*' COUNT
   (2)  478A  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
   (2)  478C  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
   (2)  478D  07                                              .BYTE   7               ;REQUEST CODE
   (1)  478E  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
   (1)  4791  3D                 4.0          DCR     A               ;DOWNCOUNT
   (1)  4792  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
   (1)  4795  F2  74  47        10.0          JP      TST020X         ;DO TEST UNTIL TILL - 0
 2723
```

```
                                .SBTTL  TEST 21 - READ CHANNEL AMTIE TEST - FROM TU PORT #3
2726  4798                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
2728  4798                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 3
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
2732  4798                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 3 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #3 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
2742  4798                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 15
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
```

```
                                        ;*RPM1 MICRO TEST 21
                                        ;*RPM1 MICRO ERROR 16
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 3 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 21
                                        ;*RPM1 MICRO ERROR 17
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 4 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 21
                                        ;*RPM1 MICRO ERROR 20
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 5 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 21
                                        ;*RPM1 MICRO ERROR 21
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 6 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 21
                                        ;*RPM1 MICRO ERROR 22
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 7 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 21
                                        ;*RPM1 MICRO ERROR 23
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL P FAILED
                                        ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
                                        ;*ACTUAL = NNNN
                                        S
                                        ; ***************************************************************************
                                        TEST21: TESTX   @021            ;INITIALIZE THE TEST
                                                        MVI     A,@021          ;DEFINE THE TEST NUMBER
                                                        CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                        ;@M8950, M8953, M8955'S
```

```
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814  4798
  1)
2815  4798
 (1)  4798   3E   11              7.0
 (1)  479A   CD   03   28        18.0
2816
2817
```

```
2818  479D  3A  41  4B      13.0  TST21X: LDA     UNITMP          ;GET THE UNIT MAP
2819  47A0  E6  08             7.0          ANI     @010            ;TEST UNIT 3?
2820  47A2  CA  C1  47        10.0          JZ      TEST22          ;NO-GO RUN TEST 22
2821  47A5  CD  A0  49        18.0          CALL    CLEAR           ;CLEAR ALL THE TU PORTS
2822  47A8  3E  03             7.0          MVI     A,@3
2823  47AA  CD  CB  49        18.0          CALL    RDCTL
2824  47AD                                   ENDTST  TST21X
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47AD                                 REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47AD  CD  06  28        18.0                                 CALL    REQST
 (2)  47B0  00                                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  47B1  00  00                                                 .WORD                   ;SYSTEM "" COUNT
 (2)  47B3  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47B5  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  47B6  07                                                     .BYTE   7               ;REQUEST CODE
 (1)  47B7  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
      47BA  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  47BB  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  47BE  F2  9D  47        10.0          JP      *ST21X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 22 - ECC SELF TEST

2826                            ST
2827                            ;*****************************************************************
2828  47C1                      ;*TEST TITLE
 (1)                            ;*---- -----
 (1)                            ;*ECC - SELF TEST
 (1)
2829                            SD
2830  47C1                      ;*****************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
 (1)                            ;*THIS TEST COMMANDS THE ERROR CORRECTION CODE MICRO CONTROLLER TO RUN
2831                            ;*ITS SELF TEST DIAGNOSTICS IN BOTH THE FORWARD AND REVERSE MODES.
2832                            SP
2833  47C1                      ;*****************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
 (1)                            ;*BGNTST
2834                            ;*     PUT THE TM78 SUBSYSTEM IN THE READ FORWARD GCR MODE
2835                            ;*     SET NORMAL WRITE AND READ CLOCKS
2836                            ;*     CALL SUBROUTINE CLEAR
2837                            ;*     CLEAR THE AMTIE LINES FROM THE USER SPECIFIED PORT
2838                            ;*     CLEAR THE DATA LINES FROM THE USER SPECIFIED PORT
2839                            ;*     DELAY 50 MICROSECONDS
2840                            ;*     ENABLE THE READ PATH
2841                            ;*     OUTPUT THE ECC SELF TEST COMMAND
2842                            ;*     DELAY 75 MICRO SECONDS
2843                            ;*     INPUT THE READ PATH STATUS
2844                            ;*       IF THE READ PATH STATUS = 'ECCOK' (ECC PASS)
2845                            ;*            THEN-CONTINUE
2846                            ;*            ELSE-ERROR
2847                            ;*       ENDIF
2848                            ;*       IF ECC STATUS=NO ROM PE,NO AMTIE,PRRT ERR,UNC,TTEC,STEC
2849                            ;*            THEN-CONTINUE
2850                            ;*            ELSE-ERROR
2851                            ;*       ENDIF
2852                            ;*     PUT THE TM78 SUBSYSTEM IN THE READ REVERSE GCR MODE
2853                            ;*     CLEAR THE AMTIE LINES FROM THE USER SPECIFIED PORT
2854                            ;*     CLEAR THE DATA LINES FROM THE USER SPECIFIED PORT
2855                            ;*     DELAY 50 MICROSECONDS
2856                            ;*     ENABLE THE READ PATH
2857                            ;*     OUTPUT THE READ CHANNEL SELF TEST COMMAND
2858                            ;*     DELAY 75 MICRO SECONDS
2859                            ;*     OUTPUT THE ECC SELF TEST COMMAND
2860                            ;*     DELAY 75 MICRO SECONDS
2861                            ;*     INPUT THE READ PATH STATUS
2862                            ;*       IF THE READ PATH STATUS = 'ECCOK' (ECC PASS)
2863                            ;*            THEN-CONTINUE
2864                            ;*            ELSE-ERROR
2865                            ;*       ENDIF
2866                            ;*       IF ECC STATUS=NO ROM PE, NO AMTIE,PTR ERR, UNC, TTEC, STEC
2867                            ;*            THEN-CONTINUE
2868                            ;*            ELSE-ERROR
2869                            ;*       ENDIF
2870
```

```
                                ;*ENDTST
 47C1                           SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 22
                                ;*RPM1 MICRO ERROR 11
                                ;*RPM1-ERROR CORRECTION-SELF TEST
                                ;*M8951, M8953, M8950
                                ;*ECC MICRO FAILED FORWARD
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*ACTUAL = ECCSTA ,EXPECTED - RSTAT
                                ;*
                                ;*RPM1 MICRO TEST 22
                                ;*RPM1 MICRO ERROR 12
                                ;*RPM1-ERROR CORRECTION-SELF TEST
                                ;*M8951, M8953, M8950
                                ;*ECC MICRO FAILED REVERSE
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*ACTUAL = ECCSTA EXPECTED = RSTAT
 47C1                           S
                                ; ***************************************************************************
```

```
2892  47C1                                    TEST22: TESTX   @22              ;INITIALIZE TEST 22
 (1)  47C1  3E  12              7.0                     MVI     A,@22           ;DEFINE THE TEST NUMBER
 (1)  47C3  CD  03  28          18.0                    CALL    TSET            ;SETUP THE TEST
2893                                    ;%RPM1-ERROR CORRECTION-SELF TEST
2894                                    ;&M8951, M8953, M8950
2895  47C6  CD  A0  49          18.0    T22ST:  CALL    CLEAR           ;CLEAR THE TU PORTS
2896  47C9  06  00                7.0            MVI     B,@0            ;LOAD THE ADDRESS OF THE BASE PORT
2897  47CB  3A  41  4B           13.0            LDA     UNITMP          ;GET THE USER SPECIFIED PORTS
2898  47CE  E6  01                7.0            ANI     $01             ;UNIT 0?
2899  47D0  C2  E6  47           10.0            JNZ     T22OUT          ;YES - EXIT UNIT SEARCH
2900  47D3  04                    4.0            INR     B               ;UPDATE THE UNIT NUMBER
2901  47D4  3A  41  4B           13.0            LDA     UNITMP          ;GET THE USER SPECIFIED PORTS
2902  47D7  E6  02                7.0            ANI     $02             ;UNIT 1?
2903  47D9  C2  E6  47           10.0            JNZ     T22OUT          ;YES - EXIT UNIT SEARCH
2904  47DC  04                    4.0            INR     B               ;UPDATE THE UNIT NUMBER
2905  47DD  3A  41  4B           13.0            LDA     UNITMP          ;GET THE USER SPECIFIED PORTS
2906  47E0  E6  04                7.0            ANI     $04             ;UNIT 2?
2907  47E2  C2  E6  47           10.0            JNZ     T22OUT          ;YES - EXIT UNIT SEARCH
2908  47E5  04                    4.0            INR     B               ;UPDATE THE UNIT NUMBER
2909  47E6  78                    4.0    T22OUT: MOV     A,B             ;SAVE THE SELECTED PORT NUMBER
2910  47E7  32  48  4B           13.0            STA     SUNIT
2911  47EA  DB  E0                10.0            IN      INTSTA          ;GET THE MASS BUS SELECT BIT
2912  47EC  E6  80                7.0            ANI     BIT7            ;SAVE THE BIT
2913  47EE  B0                    4.0            ORA     B               ;SET THE PORT BITS
2914  47EF  D3  E0                10.0           OUT     MBSEL           ;SELECT THE DESIRED TU PORT
2915  47F1  3E  10                7.0            MVI     A,W.GCR         ;PUT ENTIRE TM78 IN
2916  47F3  D3  D3                10.0           OUT     WMCCTL          ;THE WRITE GCR FORWARD MODE
2917                                                                       ;
2918  47F5  3E  08                7.0    ECCL01: MVI     A,@10           ;LOAD +10% READ CLOCK
2919  47F7  D3  F0                10.0           OUT     CLKCTL          ;SET NORMAL CLOCKS
2920  47F9  AF                    4.0            XRA     A               ;
2921  47FA  D3  48                10.0           OUT     PDIAG           ;CLEAR THE DATA LINES
2922  47FC  D3  44                10.0           OUT     TAMT            ;CLEAR THE AMTIE LINES
2923  47FE  3E  60                7.0            MVI     A,P.LCS!P.LWR
2924  4800  D3  48                10.0           OUT     PDIAG           ;JOIN THE TU PORT TO THE READ PATH
2925  4802  3E  14                7.0            MVI     A,P.WPEN!P.RPEN ;SET THE DATA PATH ENABLE BITS
2926  4804  D3  4C                10.0           OUT     PENAB
2927  4806  3E  0A                7.0            MVI     A,10            ;DELAY
2928  4808  3D                    4.0    1$:     DCR     A               ;50
2929  4809  C2  08  48           10.0            JNZ     1$              ;MICROSECONDS
2930
2931  480C  3E  0A                7.0            MVI     A,$A            ;LOAD READ PATH ENABLE BIT
2932                                                                       ;PLO DISABLE AND STATISTICS
2933  480E  D3  09                10.0           OUT     RPCTL           ;ENABLE THE READ PATH
2934                                                                       ;
2935  4810  3E  0E                7.0            MVI     A,ECCTST        ;LOAD THE ECC SELF TEST COMMAND
2936  4812  D3  08                10.0           OUT     RCMD            ;START THE TEST
2937                                    ;
2938                                    ;PERFORM A 75 MICROSECOND TIMEOUT
2939                                    ;
2940  4814  3E  0F                7.0            MVI     A,15            ;SET TIME LOOP COUNTER
2941  4816  3D                    4.0    TLOOP3: DCR     A               ;DECREMENT THE COUNTER
2942  4817  C2  16  48           10.0            JNZ     TLOOP3          ;CONTINUE-UNTIL DONE
2943                                                                       ;
```

```
 2944  481A  DB  02        10.0          IN      RSTAT           ;GET THE TEST RESULTS
 2945  481C  FE  41         7.0          CPI     ECCOK           ;ECC OK?
 2946  481E  C2  2A  48     10.0          JNZ     1$              ;NO-ERROR
 2947  4821  DB  1A        10.0          IN      ECCSTA          ;YES-CHECK THE ECC STATUS
 2948  4823  E6  6F         7.0          ANI     @157            ;JUST LOOK AT ROM PE,AMTIE,
 2949                                                            ;POINTER MISMATCH,SINGLE TRACK ERROR,
 2950                                                            ;AND DOUBLE TRACK ERROR.
 2951  4825  FE  0F         7.0          CPI     @017            ;ANY ERRORS ?
 2952  4827  CA  39  48     10.0          JZ      ECCCO1          ;NO-CONTINUE
 2953  482A  DB  1A        10.0  1$:     IN      ECCSTA          ;GET THE ECC STATUS
 2954  482C                               ROUT    ADATA           ;PUT IT IN ADATA FOR ERROR REPORT
  (1)  482C  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  482E  7F             4.0                  MOV     A,A             ;RETRY LINK
 2955  482F  DB  02        10.0          IN      RSTAT           ;GET THE READ STATUS
 2956  4831                               ROUT    EDATA           ;PUT IT IN EDATA FOR ERROR REPORT
  (1)  4831  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4833  7F             4.0                  MOV     A,A             ;RETRY LINK
 2957  4834                               ERRB    T22ST,ECCCO1
  (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  4834  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        000C                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4837  0C                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4838  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4839  CD  15  28     18.0          ECCCO1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  483C  DA  C6  47     10.0                  JC      T22ST           ;LOOP ADDRESS IF LOOP SPECIFIED
 2958                           ;>ECC MICRO FAILED FORWARD
 2959                           ;<ACTUAL DATA=ECCSTA, EXPECTED DATA=RSTAT
 2960  483F  3E  14         7.0  TST22X: MVI     A,W.GCR!W.REV
 2961  4841  D3  D3        10.0          OUT     WMCCTL
 2962                                                            ;
 2963                                                            ;
 2964  4843  3A  48  4B     13.0  ECCLOP: LDA     SUNIT           ;GET THE UNIT NUMBER
 2965  4846  47                  4.0          MOV     B,A
 2966  4847  DB  E0        10.0          IN      INTSTA          ;GET THE MASS BUS SELECT BIT
 2967  4849  E6  80         7.0          ANI     BIT7            ;SAVE THE BIT
 2968  484B  B0                  4.0          ORA     B               ;SET THE PORT BITS
 2969  484C  D3  F0        10.0          OUT     MBSEL           ;SELECT THE DESIRED TU PORT
 2970  484E  3E  08         7.0          MVI     A,@10           ;LOAD +10% READ CLOCK
 2971  4850  D3  F0        10.0          OUT     CLKCTL          ;SET NORMAL CLOCKS
 2972  4852  AF                  4.0          XRA     A               ;
 2973  4853  D3  40        10.0          OUT     TCMD            ;CLEAR THE DATA LINES
 2974  4855  D3  44        10.0          OUT     TAMT            ;CLEAR THE AMTIE LINES
 2975  4857  3E  60         7.0          MVI     A,P.LCS!P.LWR
 2976  4859  D3  48        10.0          OUT     PDIAG           ;JOIN THE TU PORT TO THE READ PATH
 2977  485B  3E  14         7.0          MVI     A,P.RPEN!P.WPEN
 2978  485D  D3  4C        10.0          OUT     PENAB
 2979  485F  3E  0A         7.0          MVI     A,10            ;DELAY
 2980  4861  3D                  4.0  1$:     DCR     A               ;50
 2981  4862  C2  61  48     10.0          JNZ     1$              ;MICROSECONDS
 2982  4865  3E  0A         7.0          MVI     A,$A            ;LOAD READ PATH ENABLE BIT
 2983                                                            ;PLO DISABLE AND STATISTICS
 2984  4867  D3  09        10.0          OUT     RPCTL           ;ENABLE THE READ PATH
 2985                                                            ;
 2986  4869  D3  08        10.0          OUT     RFIFOL          ;CLOCK THE FIFO
```

```
 2987  486B  3E  0D          7.0          MVI   A,RCLRT          ;LOAD THE READ PATH CLEAR COMMAND
 2988  486D  D3  0B         10.0          OUT   RCMD             ;ISSUE THE COMMAND
 2989                                                           ;
 2990  486F  3E  0F          7.0          MVI   A,15             ;WAIT FOR APPROX 75 MICROSEC
 2991  4871  3D              4.0  T22WT:  DCR   A
 2992  4872  C2  71  48      10.0          JNZ   T22WT            ;STAY HERE TILL READY
 2993                                                           ;
 2994  4875  3E  0E          7.0          MVI   A,ECCTST         ;LOAD THE ECC SELF TEST COMMAND
 2995  4877  D3  0B         10.0          OUT   RCMD             ;START THE ECC SELF TEST
 2996                                                           ;
 2997                                ;
 2998                                ; PERFORM A 75 MICRO SECOND TIMEOUT
 2999                                ;
 3000  4879  3E  0F          7.0          MVI   A,15             ;SET TIME LOOP COUNTER
 3001  487B  3D              4.0  TLOOP2: DCR   A                ;DECREMENT COUNTER
 3002  487C  C2  7B  48      10.0          JNZ   TLOOP2           ;CONTINUE
 3003                                                           ;
 3004                                                           ;
 3005  487F  DB  02         10.0          IN    RSTAT            ;GET THE TEST RESULTS
 3006  4881  FE  41          7.0          CPI   ECCOK            ;ECC MICRO OK?
 3007  4883  C2  8F  48      10.0          JNZ   1$               ;NO - ERROR
 3008  4886  DB  1A         10.0          IN    ECCSTA           ;YES - CHECK ECC STATUS
 3009  4888  E6  6F          7.0          ANI   @157             ;JUST LOOK AT ROM PE, AMTIE,
 3010                                                           ;POINTER MISMATCH,SINGLE TRACK ERROR,
 3011                                                           ;AND DOUBLE TRACK ERROR.
 3012  488A  FE  0F          7.0          CPI   @017             ;ANY ERRORS ?
 3013  488C  CA  9E  48      10.0          JZ    ECCCON           ;NO-CONTINUE
 3014  488F  DB  1A         10.0  1$:     IN    ECCSTA           ;GET THE ECC STATUS
 3015  4891                             ROUT  ADATA              ;PUT IT IN ADATA FOR ERROR REPORT
  (1)  4891  D3  94         10.0                OUT   ADATA              ;WRITE AC INTO ADATA
  (1)  4893  7F              4.0                MOV   A,A                ;RETRY LINK
 3016  4894  DB  02         10.0          IN    RSTAT            ;GET THE READ STATUS
 3017  4896                             ROUT  EDATA              ;PUT IT IN EDATA FOR ERROR REPORT
  (1)  4896  D3  95         10.0                OUT   EDATA              ;WRITE AC INTO EDATA
  (1)  4898  7F              4.0                MOV   A,A                ;RETRY LINK
 3018  4899                             ERRB  ECCLOP,ECCCON
  (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  4899  CD  12  28     18.0                CALL  ERLPB              ;PROCESS ERROR - DO 2.3
  (1)       000D                             MSGN  =     MSGN+1               ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  489C  0D                               .BYTE MSGN               ;MESSAGE NUMBER ID
  (1)  489D  00                               .BYTE                    ;PRINT ROUTINE NUMBER
  (1)  489E  CD  15  28     18.0          ECCCON::    CALL  CKLOP              ;CHECK LOOP FUNCTION - DO 2.2
  (1)  48A1  DA  43  48     10.0                JC    ECCLOP           ;LOOP ADDRESS IF LOOP SPECIFIED
 3019                            ;>ECC MICRO FAILED REVERSE
 3020                            ;<ACTUAL DATA=ECCSTA, EXPECTED DATA=RSTAT
 3021  48A4                             ENDTST T22ST
  (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  48A4                             REQ   7                  ;FAKE CALL TO KEEP TEST ALIVE
  (2)  48A4  CD  06  28     18.0                CALL  REQST
  (2)  48A7  00                               .BYTE                    ;DATA PATTERN NUMBER
  (2)  48A8  00  00                           .WORD                    ;SYSTEM '*' COUNT
  (2)  48AA  00  00                           .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  48AC  00                               .BYTE                    ;DATA COMPARE FLAG IF =1
  (2)  48AD  07                               .BYTE 7                  ;REQUEST CODE
```

```
   (1)  48AE  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
   (1)  48B1  3D                 4.0          DCR     A               ;DOWNCOUNT
   (1)  48B2  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
   (1)  48B5  F2  C6  47      10.0          JP      T22ST           ;DO TEST UNTIL TILL = 0
3022    48B8  CD  A0  49      18.0  EXIT:   CALL    CLEAR           ;CLEAR THE PROTS BEFORE LEAVING
3023    48BB  C3  18  28      10.0          JMP     TSTEND          ;END OF TEST
```

```
                                .SBTTL  SUBROUTINE CHECK AMTIE
                                SSUB
                                ;*************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK AMTIE SUBROUTINE
                                SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ PATH DATA
                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
                                SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
                                ;*      THEN-ERROR CHANNEL 0
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
                                ;*      THEN-ERROR CHANNEL 1
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
                                ;*      THEN-ERROR CHANNEL 2
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
                                ;*      THEN-ERROR CHANNEL 3
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
                                ;*      THEN-ERROR CHANNEL 4
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
                                ;*      THEN-ERROR CHANNEL 5
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 6 ACTUAL INFORMATION NOT - CHANNEL 6 EXPECTED
                                ;*      THEN-ERROR CHANNEL 6
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 7 ACTUAL INFORMATION NOT - CHANNEL 7 EXPECTED
                                ;*      THEN-ERROR CHANNEL 7
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL P ACTUAL INFORMATION NOT = CHANNEL P EXPECTED
                                ;*      THEN-ERROR CHANNEL P
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
```

```
3070                                        ;*ENDSUB
3071  48BE                                  S
 (1)                                        ; ***************************************************************************
3072
3073  48BE  21  46  4B      10.0  CKAMT:  LXI   H,DATAE          ;LOAD EXPECTED DATA ADDRESS
3074  48C1  3A  44  4B      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
3075  48C4                        ROUT  ADATA                    ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  48C4  D3  94           10.0                OUT   ADATA              ;WRITE AC INTO ADATA
 (1)  48C6  7F                4.0                MOV   A,A                ;RETRY LINK
3076  48C7  E6  01            7.0          ANI   $01              ;REMOVE UNWANTED BITS
3077  48C9  47                 4.0          MOV   B,A              ;SAVE IN B
3078  48CA  7E                 7.0          MOV   A,M              ;GET EXPECTED DATA
3079  48CB                        ROUT  EDATA                    ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  48CB  D3  95           10.0                OUT   EDATA              ;WRITE AC INTO EDATA
 (1)  48CD  7F                4.0                MOV   A,A                ;RETRY LINK
3080  48CE  E6  01            7.0          ANI   $01              ;REMOVE UNWANTED BITS
3081  48D0  B8                 4.0          CMP   B                ;COMPARE
3082  48D1  CA  D9  48      10.0           JZ    CKAM0            ;CONTINUE IF EQUAL
3083  48D4                    . .  ERRB    OUT,CKAM0
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  48D4  CD  12  28      18.0                CALL  ERLPB              ;PROCESS ERROR - DO 2.3
 (1)        000E                              MSGN  =     MSGN+1               ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48D7  0E                                .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  48D8  00                                .BYTE                  ;PRINT ROUTINE NUMBER
 (1)  48D9  CD  15  28      18.0        CKAM0:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48DC  DA  9F  49      10.0                JC    OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
3084                                    ;>CHANNEL 0 FAILED
3085  48DF  3A  44  4B      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
3086  48E2  E6  02            7.0          ANI   $02              ;REMOVE UNWANTED BITS
3087  48E4  47                 4.0          MOV   B,A              ;SAVE IN B
3088  48E5  7E                 7.0          MOV   A,M              ;GET EXPECTED DATA
3089  48E6  E6  02            7.0          ANI   $02              ;REMOVE UNWANTED BITS
3090  48E8  B8                 4.0          CMP   B                ;COMPARE
3091  48E9  CA  F1  48      10.0           JZ    CKAM1            ;CONTINUE IF EQUAL
3092  48EC                        ERRB    OUT,CKAM1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  48EC  CD  12  28      18.0                CALL  ERLPB              ;PROCESS ERROR - DO 2.3
 (1)        000F                              MSGN  =     MSGN+1               ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48EF  0F                                .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  48F0  00                                .BYTE                  ;PRINT ROUTINE NUMBER
 (1)  48F1  CD  15  28      18.0        CKAM1:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48F4  DA  9F  49      10.0                JC    OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
3093                                    ;>CHANNEL 1 FAILED
3094  48F7  3A  44  4B      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
3095  48FA  E6  04            7.0          ANI   $04              ;REMOVE UNWANTED BITS
3096  48FC  47                 4.0          MOV   B,A              ;SAVE IN B
3097  48FD  7E                 7.0          MOV   A,M              ;GET EXPECTED DATA
3098  48FE  E6  04            7.0          ANI   $04              ;REMOVE UNWANTED BITS
3099  4900  B8                 4.0          CMP   B                ;COMPARE
3100  4901  CA  09  49      10.0           JZ    CKAM2            ;CONTINUE IF EQUAL
```

```
3102  4904                                                    ERRB    OUT,CKAM2
 (1)                                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4904   CD   12   28        18.0                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0010                                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4907   10                                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4908   00                                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4909   CD   15   28        18.0                        CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  490C   DA   9F   49        10.0                                JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3103                                                    ;>CHANNEL 2 FAILED
3104  490F   3A   44   4B        13.0                        LDA     DATAA           ;GET THE ACTUAL DATA
3105  4912   E6   08                7.0                        ANI     $08             ;REMOVE UNWANTED BITS
3106  4914   47                     4.0                        MOV     B,A             ;SAVE IN B
3107  4915   7E                     7.0                        MOV     A,M             ;GET EXPECTED DATA
3108  4916   E6   08                7.0                        ANI     $08             ;REMOVE UNWANTED BITS
3109  4918   B8                     4.0                        CMP     B               ;COMPARE
3110  4919   CA   21   49        10.0                        JZ      CKAM3           ;CONTINUE IF EQUAL
3111  491C                                                    ERRB    OUT,CKAM3
 (1)                                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  491C   CD   12   28        18.0                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0011                                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  491F   11                                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4920   00                                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4921   CD   15   28        18.0                        CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4924   DA   9F   49        10.0                                JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3112                                                    ;>CHANNEL 3 FAILED
3113  4927   3A   44   4B        13.0                        LDA     DATAA           ;GET THE ACTUAL DATA
3114  492A   E6   10                7.0                        ANI     $10             ;REMOVE UNWANTED BITS
3115  492C   47                     4.0                        MOV     B,A             ;SAVE IN B
3116  492D   7E                     7.0                        MOV     A,M             ;GET EXPECTED DATA
3117  492E   E6   10                7.0                        ANI     $10             ;REMOVE UNWANTED BITS
3118  4930   B8                     4.0                        CMP     B               ;COMPARE
3119  4931   CA   39   49        10.0                        JZ      CKAM4           ;CONTINUE IF EQUAL
3120  4934                                                    ERRB    OUT,CKAM4
 (1)                                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4934   CD   12   28        18.0                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0012                                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4937   12                                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4938   00                                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4939   CD   15   28        18.0                        CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  493C   DA   9F   49        10.0                                JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3121                                                    ;>CHANNEL 4 FAILED
```

```
3123  493F  3A  44  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
3124  4942  E6  20      7.0          ANI     $20             ;REMOVE UNWANTED BITS
3125  4944  47          4.0          MOV     B,A             ;SAVE IN B
3126  4945  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
3127  4946  E6  20      7.0          ANI     $20             ;REMOVE UNWANTED BITS
3128  4948  B8          4.0          CMP     B               ;COMPARE
3129  4949  CA  51  49  10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
3130  494C                           ERRB    OUT,CKAM5
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  494C  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0013                               MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  494F  13                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4950  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4951  CD  15  28  18.0          CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4954  DA  9F  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3131                           ;>CHANNEL 5 FAILED
3132  4957  3A  44  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
3133  495A  E6  40      7.0          ANI     $40             ;REMOVE UNWANTED BITS
3134  495C  47          4.0          MOV     B,A             ;SAVE IN B
3135  495D  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
3136  495E  E6  40      7.0          ANI     $40             ;REMOVE UNWANTED BITS
3137  4960  B8          4.0          CMP     B               ;COMPARE
3138  4961  CA  69  49  10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
3139  4964                           ERRB    OUT,CKAM6
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4964  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0014                               MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4967  14                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4968  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4969  CD  15  28  18.0          CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  496C  DA  9F  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3140                           ;>CHANNEL 6 FAILED
3141  496F  3A  44  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
3142  4972  E6  80      7.0          ANI     $80             ;REMOVE UNWANTED BITS
3143  4974  47          4.0          MOV     B,A             ;SAVE IN B
3144  4975  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
3145  4976  E6  80      7.0          ANI     $80             ;REMOVE UNWANTED BITS
3146  4978  B8          4.0          CMP     B               ;COMPARE
3147  4979  CA  81  49  10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
3148  497C                           ERRB    OUT,CKAM7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  497C  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0015                               MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  497F  15                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4980  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4981  CD  15  28  18.0          CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4984  DA  9F  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3149                           ;>CHANNEL 7 FAILED
```

```
3151  4987  3A  45  4B      13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
3152  498A  E6  01            7.0          ANI     $01             ;REMOVE UNWANTED BITS
3153  498C  47                4.0          MOV     B,A             ;SAVE IN B
3154  498D  7E                7.0          MOV     A,M             ;GET EXPECTED DATA
3155  498E  E6  01            7.0          ANI     $01             ;REMOVE UNWANTED BITS
3156  4990  B8                4.0          CMP     B               ;COMPARE
3157  4991  CA  99  49       10.0          JZ      CKAMP           ;CONTINUE IF EQUAL
3158  4994                                  ERRA    OUT,CKAMP
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4994  CD  0F  28       18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0016                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4997  16                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4998  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4999  CD  15  28       18.0          CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  499C  DA  9F  49       10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3159                                  ;>CHANNEL P FAILED
3160                                  ;>PARITY BIT WRONG ON ACTUAL DATA BYTE
3161
3162  499F  C9                10.0  OUT:    RET
```

```
 3164                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
 3165  49A0                                  SSUB
  (1)                                        ; ****************************************************************
  (1)                                        ;*SUBROUTINE TITLE
  (1)                                        ;*---------- -----
 3166                                        ;*CLEAR ALL TU PORTS
 3167  49A0                                  SD
  (1)                                        ; ****************************************************************
  (1)                                        ;*DESCRIPTION
  (1)                                        ;*-----------
 3168                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
 3169                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
 3170                                        ;*AND LOOP MODES.
 3171  49A0                                  SP
  (1)                                        ; ****************************************************************
  (1)                                        ;*PROCEDURE
  (1)                                        ;*---------
 3172                                        ;*BGNSUB
 3173                                        ;*      SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
 3174                                        ;*      CLEAR SELECTED TRANSPORT COMMAND REGISTER A
 3175                                        ;*      CLEAR PORT SELECT FOR TRANSPORT
 3176                                        ;*      CLEAR PORT PARITY ERRORS & ENABLE WORD
 3177                                        ;*      CLEAR PORT DIAGNOSTIC CONTROL
 3178                                        ;*      CLEAR PORT AMTIE WORD
 3179                                        ;*ENDSUB
 3180  49A0                                  S
  (1)                                        ; ******************************************************************* ****
 3181  49A0  F5                      12.0    CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
 3182  49A1  C5                      12.0            PUSH    B
 3183  49A2  06    00                 7.0            MVI     B,0             ;START TO CLEAR AT PORT #0
 3184  49A4  DB    E0                10.0    CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
 3185  49A6  E6    80                 7.0            ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
 3186  49A8  B0                       4.0            ORA     B               ;ADD IN THE SELECTED PORT #
 3187  49A9  D3    E0                10.0            OUT     MBSEL           ;RESET TO THIS PORT
 3188  49AB  3E    80                 7.0            MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
 3189  49AD  D3    40                10.0            OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
 3190  49AF  AF                       4.0            XRA     A               ;CLEAR TU COMMAND A
 3191  49B0  D3    40                10.0            OUT     TCMD            ;
 3192  49B2  3E    81                 7.0            MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
 3193  49B4  D3    40                10.0            OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
 3194  49B6  3E    00                 7.0            MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
 3195  49B8  D3    40                10.0            OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
 3196  49BA  AF                       4.0            XRA     A
 3197  49BB  D3    44                10.0            OUT     TAMT            ;CLEAR AMTIE WORD
 3198  49BD  D3    48                10.0            OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
 3199  49BF  D3    4C                10.0            OUT     PENAB           ;CLEAR PORT ENABLE WORD
 3200  49C1  04                       4.0            INR     B               ;POINT TO THE NEXT PORT TO CLEAR
 3201  49C2  78                       4.0            MOV     A,B
 3202  49C3  FE    04                 7.0            CPI     4               ;DONE?
 3203  49C5  C2    A4    49          10.0            JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
 3204  49C8  C1                      10.0            POP     B               ;RESET B & C
 3205  49C9  F1                      10.0            POP     PSW             ;ALL DONE
 3206  49CA  C9                      10.0            RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE RDCTL

                                SSUB
                                ;*****************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*READ PATH CONTROL SUBROUTINE
                                SD
                                ;*****************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO CONTROL THE SEQUENCING AND LOAD OF
                                ;*AMTIE LINES TO THE SELECTED TU PORT BOARD.  DATA PATTERS ARE INCREMENTAL
                                ;*FROM 000-377.
                                SP
                                ;*****************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    SET 'DPEN'' (DATA PATH ENABLE) AND 'LWR'' (LOOP W/R) ON THE TU PORT
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CLEAR READ PATH CONTROLLER STOP BIT
                                ;*    CLEAR THE AMTIE DATA BYTE
                                ;*    BGNDO
                                ;*       :   GET THE AMTIE DATA BYTE
                                ;*       :   OUTPUT THE AMTIE DATA BYTE TO THE TU PORT AMTIE BYTE
                                ;*       :   GET THE LEAST SIGNIFICANT BIT OF THE AMTIE DATA BYTE
                                ;*       :   OUTPUT THE LEAST SIGNIFICANT BIT AS AMTIE PARITY
                                ;*       :   WAIT 50 MICRO SECONDS
                                ;*       :   INPUT THE AMTIE DATA FROM THE READ CHANNEL
                                ;*       :   INPUT THE AMTIE PARITY FROM THE READ CHANNEL
                                ;*       :   CALL ''CKAMT'' SUBROUTINE
                                ;*       :   INCREMENT THE AMTIE DATA BYTE
                                ;*       :   DO UNTIL THE AMTIE DATA BYTE = ZERO
                                ;*    ENDDO
                                ;*ENDSUB
                                S
                                ;*****************************************************************

 3237  49CB  47              4.0  RDCTL:  MOV     B,A             ;SAVE THE UNIT NUMBER
 3238  49CC  DB  E0           10.0         IN      INTSTA          ;GET THE MASS BUS PORT
 3239  49CE  E6  80            7.0         ANI     BIT7
 3240  49D0  B0                4.0         ORA     B
 3241  49D1  D3  E0           10.0         OUT     MBSEL
 3242  49D3  3E  20            7.0         MVI     A,P.LWR         ;CLEAR PORT ZERO CONTROL BYTE
 3243  49D5  D3  48           10.0         OUT     PDIAG           ;SET LOOP MODE ON THE PORT
 3244  49D7  3E  14            7.0         MVI     A,P.WPEN!P.RPEN ;SET THE DATA PATH ENABLE BITS
 3245  49D9  D3  4C           10.0         OUT     PENAB
                                                                  ;
 3247  49DB  3E  10            7.0         MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
 3248  49DD  D3  F0           10.0         OUT     CLKCTL          ;
 3249  49DF  3E  00            7.0         MVI     A,0     ;
 3250  49E1  D3  09           10.0         OUT     RPCTL
                                                                  ;
```

```
3252  49E3  AF          4.0          XRA   A          ;CLEAR THE ACCUMULATOR
3253  49E4  32 46 4B    13.0         STA   DATAE      ;STORE AS AMTIE DATA
3254                                                  ;
3255  49E7  3A 46 4B    13.0  AMTLP0: LDA   DATAE      ;GET THE AMTIE DATA
3256  49EA  47           4.0          MOV   B,A        ;SAVE IN B
3257  49EB  D3 44        10.0         OUT   TAMT       ;WRITE TO THE AMTIE LINES
3258  49ED  E6 01        7.0          ANI   $01        ;GET THE LSB
3259  49EF  32 47 4B     13.0         STA   DATAEP     ;SAVE EXPECTED DATA PARITY
3260  49F2  F6 20        7.0          ORI   P.LWR      ;
3261  49F4  D3 48        10.0         OUT   PD.AG      ;OUTPUT AS PARITY BIT
3262
3263  49F6  3E 0A        7.0          MVI   A,10       ;WAIT
3264  49F8  3D           4.0   1$:    DCR   A          ;50
3265  49F9  C2 F8 49     10.0         JNZ   1$         ;MICROSECONDS
3266                                                  ;
3267  49FC  DB 10        10.0         IN    RAMT       ;GET THE AMTIE DATA
3268  49FE  32 44 4B     13.0         STA   DATAA      ;SAVE IT
3269  4A01  DB 15        10.0         IN    RPSTA      ;GET THE AMTIE P BIT
3270  4A03  E6 01        7.0          ANI   $01        ;REMOVE EXTRA BITS
3271  4A05  32 45 4B     13.0         STA   DATAAP     ;SAVE IT
3272  4A08  CD BE 4B     18.0         CALL  CKAMT      ;GO SORT OUT ERRORS
3273  4A0B  DA E7 49     10.0         JC    AMTLP0     ;ERROR RETURN
3274  4A0E  3A 46 4B     13.0         LDA   DATAE      ;NORMAL RETURN
3275  4A11  3C           4.0          INR   A          ;UPDATE PATTERN
3276  4A12  32 46 4B     13.0         STA   DATAE      ;SAVE UPDATED PATTERN
3277  4A15  C2 E7 49     10.0         JNZ   AMTLP0     ;CONTINUE UNTIL DONE
3278  4A18  C9           10.0         RET              ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE RCCLR
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*RPM1-READ CHANNEL CLEAR SUBROUTINE
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*  SET NORMAL READ PATH CLOCK
                                        ;*  SET PLO DISABLE
                                        ;*  CLOCK THE FIFO'S
                                        ;*  ISSUE THE CLEAR ALL TEST COMMAND
                                        ;*  WAIT 40 MICROSECONDS
                                        ;*  STOP THE READ PATH AND SET PLO DISABLE
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************

4A19  3E  10          7.0  RCCLR:  MVI   A,RDCLK          ;SET NORMAL READ PATH CLOCK
4A1B  D3  F0         10.0         OUT   CLKCTL
4A1D  3E  08          7.0         MVI   A,R.PLOD         ;SET PLO DISABLE
4A1F  D3  09         10.0         OUT   RPCTL
4A21  D3  08         10.0         OUT   RFIFOL           ;CLOCK THE FIFI'S
4A23  3E  0D          7.0         MVI   A,RCLRT          ;ISSUE CLEAR ALL TEST COMMAND
4A25  D3  08         10.0         OUT   RCMD
4A27  3E  08          7.0         MVI   A,8              ;WAIT 40 MICRO SECONDS
4A29  3D              4.0  RCCLR1: DCR   A
4A2A  C2  29  4A     10.0         JNZ   RCCLR1
4A2D  3E  09          7.0         MVI   A,R.STPC!R.PLOD ;STOP THE READ PATH
4A2F  D3  09         10.0         OUT   RPCTL
4A31  C9             10.0         RET
```

```
                                .SBTTL  SUBROUTINE RCSTRT
                               SSUB
                               ;*****************************************************************************
                               ;*SUBROUTINE TITLE
                               ;*---------- -----
                               ;*RPM1-READ CHANNEL START SUBROUTINE
                               SP
                               ;*****************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNSUB
                               ;*  SET NORMAL READ PATH CLOCK
                               ;*  SET PLO DISABLE
                               ;*  CLOCK THE FIFO'S
                               ;*  ISSUE THE READ CHANNEL SELF TEST COMMAND
                               ;*  WAIT 40 MICROSECONDS
                               ;*  STOP THE READ PATH AND SET PLO DISABLE
                               ;*ENDSUB
                               S
                               ;*****************************************************************************

4A32  3E  10          7.0  RCSTRT: MVI   A,RDCLK          ;SET NORMAL READ PATH CLOCK
4A34  D3  F0         10.0         OUT   CLKCTL
4A36  3E  08          7.0         MVI   A,R.PLOD         ;SET PLO DISABLE
4A38  D3  09         10.0         OUT   RPCTL
4A3A  D3  08         10.0         OUT   RFIFOL           ;CLOCK THE FIFI'S
4A3C  3E  0C          7.0         MVI   A,RCHTST         ;ISSUE READ CHANNEL SELF TEST COMMAND
4A3E  D3  08         10.0         OUT   RCMD
4A40  3E  08          7.0         MVI   A,8              ;WAIT 40 MICRO SECONDS
4A42  3D              4.0  RCSTR1: DCR   A
4A43  C2  42  4A     10.0         JNZ   RCSTR1
4A46  3E  09          7.0         MVI   A,R.STPC!R.PLOD ;STOP THE READ PATH
4A48  D3  09         10.0         OUT   RPCTL
4A4A  C9             10.0         RET
```

```
                                .SBTTL  SUBROUTINE RCCK
                                SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*RPM1-READ CHANNEL SELF TEST CHECK
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*  GET THE READ PATH END MARK REGISTER
                                ;*  IF END MARK REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH DATA REGISTER
                                ;*  IF DATA REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH POSTAMBLE REGISTER
                                ;*  IF POSTAMBLE REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH ILLEGAL REGISTER
                                ;*  IF ILLEGAL REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH MARK TWO REGISTER
                                ;*  IF MARK TWO REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH NOT DONE REGISTER
                                ;*  IF NOT DONE REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDSUB
                                S
                                ; ***************************************************************************

                    10.0  RCCK:   IN      REND            ;GET THE END BITS
4A4D  32  4C  4B    13.0          STA     XEND            ;SAVE
4A50  DB  17        10.0          IN      RDATA           ;GET THE DATA BITS
4A52  32  4D  4B    13.0          STA     XDATA           ;SAVE
4A55  DB  16        10.0          IN      RPOSTN          ;GET THE POSTAMBLE BITS
4A57  32  4E  4B    13.0          STA     XPOSTN          ;SAVE
4A5A  DB  12        10.0          IN      RILL            ;GET THE ILLEGAL BITS
4A5C  32  4F  4B    13.0          STA     XILL            ;SAVE
4A5F  DB  13        10.0          IN      RMK2            ;GET THE MARK 2 BITS
```

```
3385  4A61  32  50  4B      13.0          STA   XMK2          ;SAVE
3386  4A64  DB  11          10.0          IN    RDON          ;GET THE DONE BITS
3387  4A66  32  51  4B      13.0          STA   XDON          ;SAVE
3388  4A69  C3  B7  4A      10.0          JMP   RCCKX         ;CONTINUE
3389
3390  4A6C  DB  15          10.0  RCCKP:  IN    RPSTA         ;GET THE END PARITY BIT
3391  4A6E  E6  10           7.0          ANI   R.END         ;
3392  4A70  CA  75  4A      10.0          JZ    RCCKP1        ;
3393  4A73  3E  01           7.0          MVI   A,1           ;
3394  4A75  32  4C  4B      13.0  RCCKP1: STA   XEND          ;STORE THE END PARITY BIT RIGHT JUSTIFIED
3395  4A78  DB  15          10.0          IN    RPSTA         ;GET THE DATA PARITY BIT
3396  4A7A  E6  40           7.0          ANI   R.DATA        ;
3397  4A7C  CA  81  4A      10.0          JZ    RCCKP2        ;
3398  4A7F  3E  01           7.0          MVI   A,1           ;
3399  4A81  32  4D  4B      13.0  RCCKP2: STA   XDATA         ;STORE THE DATA PARITY BIT RIGHT JUSTIFIED
3400  4A84  DB  15          10.0          IN    RPSTA         ;GET THE POSTAMBLE PARITY BIT
3401  4A86  E6  20           7.0          ANI   R.POST        ;
3402  4A88  CA  8D  4A      10.0          JZ    RCCKP3        ;
3403  4A8B  3E  01           7.0          MVI   A,1           ;
3404  4A8D  32  4E  4B      13.0  RCCKP3: STA   XPOSTN        ;STORE THE POSTAMBLE PARITY BIT RIGHT JUSTIFIED
3405  4A90  DB  15          10.0          IN    RPSTA         ;GET THE ILLEGAL PARITY BIT
3406  4A92  E6  04           7.0          ANI   R.ILL         ;
3407  4A94  CA  99  4A      10.0          JZ    RCCKP4        ;
3408  4A97  3E  01           7.0          MVI   A,1           ;
3409  4A99  32  4F  4B      13.0  RCCKP4: STA   XILL          ;STORE THE ILLEGAL PARITY BIT RIGHT JUSTIFIED
3410  4A9C  DB  15          10.0          IN    RPSTA         ;GET THE MARK 2 PARITY BIT
3411  4A9E  E6  08           7.0          ANI   R.MK2         ;
3412  4AA0  CA  A5  4A      10.0          JZ    RCCKP5        ;
3413  4AA3  3E  01           7.0          MVI   A,1           ;
3414  4AA5  32  50  4B      13.0  RCCKP5: STA   XMK2          ;STORE THE MARK2 PARITY BIT RIGHT JUSTIFIED
3415  4AA8  DB  15          10.0          IN    RPSTA         ;GET THE DONE PARITY BIT
3416  4AAA  E6  02           7.0          ANI   R.DON         ;
3417  4AAC  CA  B1  4A      10.0          JZ    RCCKP6        ;
3418  4AAF  3E  01           7.0          MVI   A,1           ;
3419  4AB1  32  51  4B      13.0  RCCKP6: STA   XDON          ;STORE THE DONE PARITY BIT RIGHT JUSTIFIED
3420  4AB4  C3  B7  4A      10.0          JMP   RCCKX
3421
3422  4AB7  3A  4B  4B      13.0  RCCKX:  LDA   RCMSK          ;GET THE CHANNEL MASK
3423  4ABA  4F               4.0          MOV   C,A           ;SAVE IN C
3424  4ABB  3A  4A  4B      13.0          LDA   RCEDAT        ;GET THE EXPECTED DATA
3425  4ABE  47               4.0          MOV   B,A           ;SAVE IN B
3426  4ABF  A7               4.0          ANA   A             ;SET CONDITION BITS
3427  4AC0  C2  C9  4A      10.0          JNZ   RCCK1         ;GO PROCESS NOT 0
3428  4AC3                                ROUT  EDATA         ;ZERO
 (1)  4AC3  D3  95          10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  4AC5  7F               4.0                MOV   A,A             ;RETRY LINK
3429  4AC6  C3  CE  4A      10.0          JMP   RCCK2
3430  4AC9  3E  01           7.0  RCCK1:  MVI   A,1
3431  4ACB                                ROUT  EDATA
 (1)  4ACB  D3  95          10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  4ACD  7F               4.0                MOV   A,A             ;RETRY LINK
3432  4ACE  3A  4C  4B      13.0  RCCK2:  LDA   XEND          ;GET THE END MARK REGISTER
3433  4AD1  A1               4.0          ANA   C             ;AND WITH CHANNEL MASK
3434  4AD2  B8               4.0          CMP   B             ;IF RESULT=EXPECTED DATA
```

```
3435  4AD3  CA  DB  4A  10.0          JZ      RCCN1           ;THEN-CONTINUE
3436  4AD6                           ERRE    RCOUT,RCCN1
 (1)                                 ;FLAG ERROR - WITH EXPECTED DATA 'EDATA' VALID
 (1)  4AD6  CD  0C  28  18.0                    CALL    ERLPE
 (1)       0017                                MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4AD9  17                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4ADA  00                                  .BYTE
 (1)  4ADB  CD  15  28  18.0            RCCN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4ADE  DA  40  4B  10.0                    JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3437                            ;>END MARK BIT INCORRECT
3438  4AE1  3A  4D  4B  13.0          LDA     XDATA           ;GET THE DATA REGISTER
3439  4AE4  A1                 4.0    ANA     C               ;AND WITH CHANNEL MASK
3440  4AE5  B8                 4.0    CMP     B               ;IF RESULT=EXPECTED DATA
3441  4AE6  CA  EE  4A  10.0          JZ      RCCN2           ;THEN-CONTINUE
3442  4AE9                           ERRE    RCOUT,RCCN2
 (1)                                 ;FLAG ERROR - WITH EXPECTED DATA 'EDATA' VALID
 (1)  4AE9  CD  0C  28  18.0                    CALL    ERLPE
 (1)       0018                                MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4AEC  18                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4AED  00                                  .BYTE
 (1)  4AEE  CD  15  28  18.0            RCCN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4AF1  DA  40  4B  10.0                    JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3443                            ;>DATA BIT INCORRECT
3444  4AF4  3A  4E  4B  13.0          LDA     XPOSTN          ;GET THE POSTAMBLE REGISTER
3445  4AF7  A1                 4.0    ANA     C               ;AND WIITH CHANNEL MASK
3446  4AF8  B8                 4.0    CMP     B               ;IF RESULT=EXPECTED DATA
3447  4AF9  CA  01  4B  10.0          JZ      RCCN3           ;THEN-CONTINUE
3448  4AFC                           ERRE    RCOUT,RCCN3
 (1)                                 ;FLAG ERROR - WITH EXPECTED DATA 'EDATA' VALID
 (1)  4AFC  CD  0C  28  18.0                    CALL    ERLPE
 (1)       0019                                MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4AFF  19                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4B00  00                                  .BYTE
 (1)  4B01  CD  15  28  18.0            RCCN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4B04  DA  40  4B  10.0                    JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3449                            ;>POSTAMBLE BIT INCORRECT
3450  4B07  3A  4F  4B  13.0          LDA     XILL            ;GET THE ILLEGAL REGISTER
3451  4B0A  A1                 4.0    ANA     C               ;AND WITH CHANNEL MASK
3452  4B0B  B8                 4.0    CMP     B               ;IF RESULT=EXPECTED DATA
3453  4B0C  CA  14  4B  10.0          JZ      RCCN4           ;THEN-CONTINUE
3454  4B0F                           ERRE    RCOUT,RCCN4
 (1)                                 ;FLAG ERROR - WITH EXPECTED DATA 'EDATA' VALID
 (1)  4B0F  CD  0C  28  18.0                    CALL    ERLPE
 (1)       001A                                MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4B12  1A                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4B13  00                                  .BYTE
 (1)  4B14  CD  15  28  18.0            RCCN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4B17  DA  40  4B  10.0                    JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3455                            ;>ILLEGAL BIT INCORRECT
3456  4B1A  3A  50  4B  13.0          LDA     XMK2            ;GET THE MARK TWO REGISTER
3457  4B1D  A1                 4.0    ANA     C               ;AND WITH CHANNEL MASK
3458  4B1E  B8                 4.0    CMP     B               ;IF RESULT=EXPECTED DATA
3459  4B1F  CA  27  4B  10.0          JZ      RCCN5           ;THEN-CONTINUE
```

```
                                                ERRE    RCOUT,RCCN5
                                                ;FLAG ERROR - WITH EXPECTED DATA 'EDATA' VALID
                                                        CALL    ERLPE
                                                        MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE
                                                RCCN5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>MARK TWO BIT INCORRECT
                                                LDA     XDON            ;GET THE NOT DONE REGISTER
                                                ANA     C               ;AND WITH CHANNEL MASK
                                                CMP     B               ;IF RESULT=EXPECTED DATA
                                                JZ      RCCN6           ;THEN-CONTINUE
                                                ERRE    RCOUT,RCCN6
                                                ;FLAG ERROR - WITH EXPECTED DATA 'EDATA' VALID
                                                        CALL    ERLPE
                                                        MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE
                                                RCCN6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                        JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>DONE NOT BIT INCORRECT
4B40  C9                        10.0    RCOUT:  RET
```

```
                                        .SBTTL  PROGRAM VARIABLES
3471
3472  4B41    00                        UNITMP: .BYTE   0               ;UNIT MAP BYTE
3473  4B42    00                        RUNSTA: .BYTE   0               ;EXPECTED RUN TIME STATUS BYTE
3474  4B43    00                        RUNCMD: .BYTE   0               ;RUN TIME COMMAND BYTE
3475  4B44    00                        DATAA:  .BYTE   0               ;DATA BYTE ACTUAL
3476  4B45    00                        DATAAP: .BYTE   0               ;DATA BYTE ACTUAL PARITY
3477  4B46    00                        DATAE:  .BYTE   0               ;DATA BYTE EXPECTED
3478  4B47    00                        DATAEP: .BYTE   0               ;DATA BYTE EXPECTED PARITY
3479  4B48    00                        SUNIT:  .BYTE   0               ;TAPE UNIT NUMBER
3480  4B49    00                        JAMDAT: .BYTE   0               ;TIE BUS JAM VALUE
3481  4B4A    00                        RCEDAT: .BYTE   0               ;EXPECTED READ CHANNEL DATA
3482  4B4B    00                        RCMSK:  .BYTE   0               ;READ CHANNEL MASK
3483  4B4C    00                        XEND:   .BYTE   0               ;COPY OF REND REGISTER
3484  4B4D    00                        XDATA:  .BYTE   0               ;COPY OF RDATA REGISTER
3485  4B4E    00                        XPOSTN: .BYTE   0               ;COPY OF RPOSTN REGISTER
3486  4B4F    00                        XILL:   .BYTE   0               ;COPY OF RILL REGISTER
3487  4B50    00                        XMK2:   .BYTE   0               ;COPY OF RMK2 REGISTER
3488  4B51    00                        XDON:   .BYTE   0               ;COPY OF RDON REGISTER
3489          0000                              .END
```

RPM1.M80 SYMBOL TABLE

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTLPO  49E7           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTFH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B         CBYTL  = 008A          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093         CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D         CDVTL  = 008C          CHPTIE= 0028
CH0TIE= 0020           CH1TIE= 0021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKAMP   4999  G        CKAMT   48BE          CKAM0   48D9  G       CKAM1   48F1  G
CKAM2   4909  G        CKAM3   4921  G       CKAM4   4939  G       CKAM5   4951  G
CKAM6   4969  G        CLOCK   4F26          CKLOP  = 2815          CLEAR   49A0
CLKCTL= 00F0           CMC1H  = 009B         CLRLP   49A4          CMC0H  = 0099
CMC0L  = 0098          CMC3H  = 009F         CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CNTCTL= 00D7          CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CSRLL  = 0090         CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CTSTL  = 008E         CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CXINL  = 0082         CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          C.DSE  = 0010         C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.FMT  = 0070         C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.MAIN= 0020          C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.SHR  = 0040         C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          D      =%0002         C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          DATAE   4B46          DATAA   4B44          DATAAP  4B45
DATACT= 00D0           DBUSST= 00C0          DATAEP  4B47          DBUS    4F28
DBUSCT= 00C0           DDRBIN= 0002          DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRCTL= 00DB          DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DIARD  = 0008         DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DSE    = 0006         DONE1  = 0045          DONINT= 0010
DSAVE   4F9E           D.EOTD= 0010          DUMMY   4719  G       D.ATH0= 0001
D.ATH1= 0002           D.TACH= 0008          D.LAGC= 0020          D.NOTW= 0040
D.NTHR= 0004           ECCCON  489E  G       D.WR4  = 0080          E      =%0003
ECCBAD= 0042           ECCLO1  47F5          ECCCOR= 0019          ECCCO1  4839  G
ECCLOP  4843           EDATA  = 0095         ECCOK  = 0041          ECCSTA= 001A
ECCTST= 000E           ERLPA  = 280F         EOTCLR= 0003          ERFLG   4F93
ERLP   = 2809          ERRCNT= 00D6          ERLPB  = 2812          ERLPE  = 280C
ERNUM   4F90           E.AMT  = 0020         ESAVE   4F9F          EXIT    48B8
E.ACRC= 0010           E.RPE  = 0040         E.CDP  = 0080          E.CRC  = 0080
E.PNTR= 0008           FIFORD= 006A          E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004          GCRSET= 0002          FORMAT  4F25          FWDTST= 0061
GCRID  = 0089          IE     = 0008         GOODTM= 0092          H      =%0004
HLSAVE  4FA0           I.RMPE= 0040          INTSTA= 00E0          ITERA   4F9A
I.PWR  = 0020          JAMDAT  4B49          I5.5   = 0010          I6.5   = 0020
I7.5   = 003F          KENAB  = 0078         KCALL  = 005F          KCLR   = 007B
KDEP   = 003F          KEY10  = 006D         KEXAM  = 003E          KEYBRD= 00C8
KEY1   = 0078          KEY14  = 005D         KEY11  = 006E          KEY12  = 006F
KEY13  = 005C          KEY18  = 003D         KEY15  = 005E          KEY16  = 005F
KEY17  = 003C          KEY3   = 007A         KEY19  = 003E          KEY2   = 0079
KEY20  = 003F                                KEY4   = 007B          KEY5   = 0074
```

```
KEY6   = 0075    KEY7   = 0076    KEY8   = 0077    KEY9   = 006C
KINTA  = 006F    KLDAD  = 003D    KN0    = 003C    KN1    = 005C
KN2    = 005D    KN3    = 005E    KN4    = 006C    KN5    = 006D
KN6    = 006E    KN7    = 0074    KN8    = 0075    KN9    = 0076
KU2    = 0079    KU3    = 007A    KU8    = 0077    L      =%0005
LBLANK= 00CF     LCE    = 0008    LCH    = 000C    LCL    = 000D
LCP    = 000E    LC0    = 0000    LC1    = 0001    LC2    = 0002
LC3    = 0003    LC4    = 0004    LC5    = 0005    LC6    = 0006
LC7    = 0007    LC8    = 0008    LC9    = 0009    LDLEDA= 00CA
LDLEDB= 00CB     LDLEDC= 00CC     LDLEDD= 00CD     LDLEDE= 00CE
LDLEDF= 00CF     LKDIAG= 2800     LKKBD  = 004C    LKKEY  = 0049
LKLWMG= 0052     LKLWMP= 0055     LKLWPG= 0052     LKLWPP= 004F
LKMOD7= 0046     LKOPR  = 0046    LPFLG    4F94    LPNUM    4F92
M      -%0006    MBSEL  = 00E0    MB.A   = 0008    MB.B   = 0004
MEMTOP= 4FFF     MINUS  = 000A    MM     = 8000    MSE    = 0008
MSGN   - 001C    MTACLR= 0000     MT.ARA= 0020     MT.CPE= 0080
MT.DSE= 0001     MT.FWD= 0040     MT.INH= 0008     MT.LWR= 0004
MT.MOT= 0002     MT.NWT- 0080     MT.PEC= 0040     MT.PSB= 0004
MT.PSO= 0001     MT.RST= 0002     MT.REV= 0020     MT.WRT= 0010
MT.Z   - 0008    M.ATA  = 0080    M.CAPE= 0020     M.CONT= 0080
M.DEM  - 0020    M.EBL  = 0010    M.EXC  = 0008    M.FAIL= 0008
M.ILR  = 0010    M.INIT- 0010     M.OCC  = 0020    M.ONLI= 0001
M.PE   - 0040    M.PORT= 0080     M.RDEN= 0002     M.RDPE= 0008
M.RUN  = 0004    M.SCLK= 0001     M.TRA  = 0040    M.UNIT= 0007
M.WCLK- 0040     M.WCLN= 0080     M.WREN= 0080     M5.5   = 0001
M6.5   - 0002    M7.5   = 0004    NOTCAP= 0088     OKAY   = 00FF
OPRRAM- 4300     OPSTRT= 0058     OPVER  = 0040    OUT      499F
PADCNT= 00D5     PADCRC= 0080     PDIAG  = 0048    PEID   = 008A
PENAB  - 004C    PESET  = 0001    PL     - 00B1    PRDD   = 004C
PRENF  - 009C    PS     = 00B2    PSTAT  = 0048    PSW    =%0009
P.AMTP   0001    P.BCTC- 0040     P.CMDP= 0020     P.INTE= 0080
P.LCS  - 0040    P.LWR  = 0020    P.RDP  = 0002    P.RPEN= 0004
P.RPSI= 0002     P.RPOE= 0020     P.RP1E= 0010     P.RP2E= 0020
P.RP3E- 0010     P.SING= 0080     P.STAT= 0002     P.STPE= 0080
P.TACH- 0008     P.TUPR= 0010     P.WCSP= 0004     P.WDS  = 0040
P.WFLP= 0001     P.WPEN= 0010     P.WPOE= 0008     P.WP1E= 0004
P.WP2E= 0008     P.WP3E= 0004     P.5VOK= 0002     QUE    = 281B
QUEM   - 281E    RAMT   = 0010    RARA   = 0006    RARAI  = 0004
RCCK     4A4B    RCCKP    4A6C    RCCKP1   4A75    RCCKP2   4A81
RCCKP3   4A8D    RCCKP4   4A99    RCCKP5   4AA5    RCCKP6   4AB1
RCCKX    4AB7    RCCK1    4AC9    RCCK2    4ACE    RCCLR    4A19
RCCLR1   4A29    RCCN1    4ADB  G RCCN2    4AEE  G RCCN3    4B01  G
RCCN4    4B14  G RCCN5    4B27  G RCCN6    4B3A  G RCEDAT   4B4A
RCHBDO- 0048     RCHBD1= 0047     RCHOK  = 0046    RCHTST= 000C
RCLRT  = 000D    RCMD   = 0008    RCMLP  = 0003    RCMSK    4B4B
RCONT  = 0080    RCOUT    4B40    RCSTRT   4A32    RCSTR1   4A42
RDATA  = 0017    RDCLK  = 0010    RDCTL    49CB    RDON   = 0011
READG  = 0007    REND   = 0014    REQST  = 2806    RESCHR= 00D1
RETURN   4454    REVTST= 0064     REWIND= 0004     RFIFOL= 0008
RGCLK  - 0002    RGCRI  - 0003    RIBG   = 0001    RILL   = 0012
RINST  = 000C    RLOOP1   44BD    RMCCN0   4378  G RMCCN1   4478  G
RMCCN2   4494  G RMCCN3   44B7  G RMCCN4   44D7  G RMCLP0   4360
RMCLP1   4462    RMCTST   0008    RMK2   = 0013    RNOP   = 0000
RPATH  = 0001    RPBAD  - 0044    RPCHI  = 0001    RPCLK  = 0003
```

```
RPCTL = 0009        RPEI  = 0002        RPFAIL= 0000        RPF1  = 009D
RPF2  = 009E        RPOK  = 0043        RPOSTN= 0016        RPSTA = 0015
RRCMT = 000A        RSTAT = 0002        RTIEB = 000A        RTIER = 0030
RTM   = 0005        RUNCMD  4B43        RUNKI = 0009        RUNSTA  4B42
RUPTST= 005E        RWDUNL= 0005        R.AMT = 0001        R.BOP = 0008
R.DATA= 0040        R.DON = 0002        R.DRDY= 0010        R.END = 0010
R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008        R.PLOD= 0008
R.PL00= 0010        R.PL01= 0020        R.POST= 0020        R.STNM= 0002
R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040
R.VOK = 0080        R00H  = 0081        R00L  = 0080        R01H  = 0083
R01L  = 0082        R02H  = 0085        R02L  = 0084        R03H  = 0087
R03L  = 0086        R04H  = 0089        R04L  = 0088        R05H  = 008B
R05L  = 008A        R06H  = 008D        R06L  = 008C        R07H  = 008F
R07L  = 008E        R10H  = 0091        R10L  = 0090        R11H  = 0093
R11L  = 0092        R12H  = 0095        R12L  = 0094        R13H  = 0097
R13L  = 0096        R14H  = 0099        R14L  = 0098        R15H  = 009B
R15L  = 009A        R16H  = 009D        R16L  = 009C        R17H  = 009F
R17L  = 009E        R7.5  = 0010        SELCLR= 0000        SETATA= 00A1
SID   = 0080        SOD   = 0080        SOE   = 0040        SP    =%0008
SSCLK = 0040        SSTEP = 0005        STACK = 4FFF        STACN1  443F
STALP1  4420        STATCK  4412        STATRM= 4F20        STPCT   4F20
STRSP = 5000        SUNIT   4B48        TADR00= 0080        TADR01= 0081
TADR02= 0082        TADR03= 0083        TADR04= 0084        TADR05= 0085
TADR06= 0086        TADR07= 0087        TADR10= 0088        TADR11= 0089
TADR12= 008A        TADR13= 008B        TAMT  = 0044        TASEL = 0080
TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004
TC.REV= 0020        TC.WRT= 0010        TEMP    4F99        TEST05  44F1
TEST06  452B        TEST07  4565        TEST1   4300        TEST10  459F
TEST11  45D9        TEST12  4613        TEST13  464D        TEST14  4687
TEST15  46C1        TEST16  46FB        TEST17  4746        TEST2   434F
TEST20  476F        TEST21  4798        TEST22  47C1        TEST3   439D
TEST4   4455        TIECN0  4329  G     TIEL0   4311        TLOOP0  4468
TLOOP2  487B        TLOOP3  4816        TLOOP4  44C7        TMR   = 0099
TMRDY = 0040        TRKENA= 00D2        TSET  = 2803        TSTEND= 2818
TSTS  = 0040        TST01X  4305        TST02X  4354        TST020  4774
TST03A  43A3        TST03B  43B2  G     TST03C  43C8  G     TST03D  43DE  G
TST03E  43F4  G     TST03X  43A2        TST04X  445A        TST05X  44F6
TST05Y  4509        TST06X  4530        TST06Y  4543        TST07X  456A
TST07Y  457D        TST10X  45A4        TST10Y  45B7        TST11X  45DE
TST11Y  45F1        TST12X  4618        TST12Y  462B        TST13X  4652
TST13Y  4665        TST14X  468C        TST14Y  469F        TST15X  46C6
TST15Y  46D9        TST16X  4722        TST17X  474B        TST21X  479D
TST22X  483F        TUSEL0= 00D1        TUSEL1= 00D2        TU78  = 0010
T.ATH0  0001        T.ATH1= 0002        T.BOT = 0004        T.EOT = 0002
T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020        T.PES = 0008
T.PSBJ= 0020        T.PSOJ= 0008        T.PS1J= 0010        T.RDY = 0080
T.RDYO= 0040        T.RWD = 0010        T.SCLK= 0002        T220UT  47E6
T22ST   47C6        T22WT   4871        UIBG  = 00A1        UNITMP  4B41
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTLAT= 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
```

```
W.SKIP= 000F            W.WRIT= 0008            W.XFER= 0020            X     =%000A
XDATA   4B4D            XDON    4B51            XEND    4B4C            XILL    4B4F
XMK2    4B50            XPOSTN  4B4E            X.DONN= 0080            X.ENAB= 0040
X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001            Y     =%000B
.     = 4B52

ERRORS DETECTED:  0

*RPM1.A78/PTP,RPM1-NLIST,PARAM,MACRO,LIST,RPM1
RUN-TIME: 6 9 0 SECONDS
CORE USED:  10K
```

```
                                .TITLE  RPM2 - READ PATH MICROCONTROLLER PART #2
                                .SBTTL  TEST 01 - READ CHANNEL FIFO STATIC TEST/ALL ONES
                                ;ID RPM2-READ PATH MICRO CONTROLLER PART #2

4300                            ST
                                ; **********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*READ CHANNEL - FIFO STATIC TEST - ALL ONES
4300                            SD
                                ; **********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE FIFO'S IN PLO BYPASS MODE 3 WHERE ALL 9 FIFO'S ARE
                                ;*PARALLEL LOADED WITH ONES FROM THE READ CHANNEL TEST DATA BIT.  ONCE
                                ;*LOADED THE READ PATH RUNS A SPECIAL TEST ROUTINE TO READ ONE BIT FROM
                                ;*EACH FIFO AND LOAD IT IN THE DATA REGISTER.  THEN THIS REGISTER IS
                                ;*COMPARED WITH THE VALUE WRITTEN AND ANY ERRORS ARE REPORTED.  THE FAILING
                                ;*M8950 MODULE CAN BE LOCATED BY THE BIT POSITION OF THE FAILING DATA BIT(S).
4300                            SP
                                ; **********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    SET PLO BYPASS MODE 3
                                ;*    CLOCK THE FIFO'S
                                ;*    ISSUE THE CLEAR ALL FIFO'S COMMAND
                                ;*    DELAY 5 MICRO SECONDS
                                ;*    STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*    SET THE FIFO TEST DATA BIT
                                ;*    CLOCK THE TEST DATA INTO THE FIFO'S
                                ;*    SET THE SUBSYSTEM IN THE PE READ MODE
                                ;*    ISSUE THE FIFO READ COMMAND
                                ;*    ENABLE THE READ PATH AND READ CHANNELS CLOCK
                                ;*    DELAY 5 MICRO SECONDS
                                ;*    STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*    READ THE FIFO DATA (9 BITS)
                                ;*    SET UP THE EXPECTED DATA OF ALL ONES
                                ;*    CALL SUBROUTINE CKAMT
                                ;*ENDTST
4300                            SE
                                ; **********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 01
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-FIFO STATUS TEST-ALL ONES
                                ;*M8950, M8953
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 01
```

```
1371                                    ;*RPM2 MICRO ERROR 10
1372                                    ;*RPM2-FIFO STATUS TEST-ALL ONES
1373                                    ;*M8950, M8953
1374                                    ;*CHANNEL 1 FAILED
1375                                    ;*ACTUAL = NNNN
1376                                    ;*EXPECTED = NNNN
1377                                    ;*
1378                                    ;*RPM2 MICRO TEST 01
1379                                    ;*RPM2 MICRO ERROR 11
1380                                    ;*RPM2-FIFO STATUS TEST-ALL ONES
1381                                    ;*M8950, M8953
1382                                    ;*CHANNEL 2 FAILED
1383                                    ;*ACTUAL = NNNN
1384                                    ;*EXPECTED = NNNN
1385                                    ;*
1386                                    ;*RPM2 MICRO TEST 01
1387                                    ;*RPM2 MICRO ERROR 12
1388                                    ;*RPM2-FIFO STATUS TEST-ALL ONES
1389                                    ;*M8950, M8953
1390                                    ;*CHANNEL 3 FAILED
1391                                    ;*ACTUAL = NNNN
1392                                    ;*EXPECTED = NNNN
1393                                    ;*
1394                                    ;*RPM2 MICRO TEST 01
1395                                    ;*RPM2 MICRO ERROR 13
1396                                    ;*RPM2-FIFO STATUS TEST-ALL ONES
1397                                    ;*M8950, M8953
1398                                    ;*CHANNEL 4 FAILED
1399                                    ;*ACTUAL = NNNN
1400                                    ;*EXPECTED = NNNN
1401                                    ;*
1402                                    ;*RPM2 MICRO TEST 01
1403                                    ;*RPM2 MICRO ERROR 14
1404                                    ;*RPM2-FIFO STATUS TEST-ALL ONES
1405                                    ;*M8950, M8953
1406                                    ;*CHANNEL 5 FAILED
1407                                    ;*ACTUAL = NNNN
1408                                    ;*EXPECTED = NNNN
1409                                    ;*
1410                                    ;*RPM2 MICRO TEST 01
1411                                    ;*RPM2 MICRO ERROR 15
1412                                    ;*RPM2-FIFO STATUS TEST-ALL ONES
1413                                    ;*M8950, M8953
1414                                    ;*CHANNEL 6 FAILED
1415                                    ;*ACTUAL = NNNN
1416                                    ;*EXPECTED = NNNN
1417                                    ;*
1418                                    ;*RPM2 MICRO TEST 01
1419                                    ;*RPM2 MICRO ERROR 16
1420                                    ;*RPM2-FIFO STATUS TEST-ALL ONES
1421                                    ;*M8950, M8953
1422                                    ;*CHANNEL 7 FAILED
1423                                    ;*ACTUAL = NNNN
1424                                    ;*EXPECTED - NNNN
```

```
                                        ;*
                                        ;*RPM2 MICRO TEST 01
                                        ;*RPM2 MICRO ERROR 17
                                        ;*RPM2-FIFO STATUS TEST-ALL ONES
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER - LLL
                                        $
4300                                    ; ************************************************************************************
 (1)
                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
4300  3E  01            7.0                         MVI     A,@1          ;DEFINE THE TEST NUMBER
4302  CD  03  28        18.0                        CALL    TSET          ;SETUP THE TEST
                                        ;%RPM2-FIFO STATIC TEST-ALL ONES
                                        ;&M8950, M8953
4305  3E  10            7.0   TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
4307  D3  F0           10.0          OUT     CLKCTL          ;
4309  3E  30            7.0   FIFOL1: MVI     A,R.PLO1!R.PLO0 ;SET PLO BYPASS MODE 3
430B  D3  09           10.0          OUT     RPCTL           ;
430D  D3  08           10.0          OUT     RFIFOL          ;CLOCK THE FIFO
430F  3E  0D            7.0          MVI     A,RCLRT         ;LOAD ALL FIFO'S CLEAR COMMAND
4311  D3  08           10.0          OUT     RCMD            ;ISSUE FIFO CLEAR
4313  00              4.0           NOP                     ;WAIT
4314  00              4.0           NOP                     ;WAIT
4315  00              4.0           NOP                     ;WAIT
4316  00              4.0           NOP                     ;WAIT
4317  3E  71            7.0         MVI     A,R.PLO1!R.PLO0!R.STPC!@100
4319  D3  09           10.0         OUT     RPCTL           ;STOP THE RMC WITH FIFO DATA
                                                             ;OF ONES
431B  D3  08           10.0         OUT     RFIFOL          ;CLOCK ''1'S'' INTO THE FIFO
431D  AF                4.0         XRA     A               ;
431E  D3  D3           10.0         OUT     WMCCTL          ;PUT MACHINE IN PE MODE
4320  3E  6A            7.0         MVI     A,FIFORD        ;LOAD THE FIFO READ COMMAND
4322  D3  0B           10.0         OUT     RCMD            ;
4324  3E  30            7.0         MVI     A,R.PLO0!R.PLO1 ;RUN THE RMC
4326  D3  09           10.0         OUT     RPCTL           ;
4328  00              4.0           NOP                     ;WAIT
4329  00              4.0           NOP                     ;WAIT
432A  00              4.0           NOP                     ;WAIT
432B  00              4.0           NOP                     ;WAIT
432C  3E  31            7.0         MVI     A,R.PLO0!R.PLO1!R.STPC  ;STOP THE RMC
432E  D3  09           10.0         OUT     RPCTL
4330  DB  17           10.0         IN      RDATA           ;READ THE FIFO DATA
4332  32  45  48       13.0         STA     DATAA           ;STORE ACTUAL DATA
4335  3E  FF            7.0         MVI     A,$FF           ;LOAD EXPECTED DATA
4337  32  43  48       13.0         STA     DATAE           ;STORE EXPECTED DATA
433A  DB  15           10.0         IN      RPSTA           ;GET PARITY BITS
433C  E6  40            7.0         ANI     R.DATA          ;GET DATA PARITY
433E  07                4.0         RLC                     ;JUSTIFY THE
433F  07                4.0         RLC                     ;ACTUAL PARITY DATA
4340  32  46  48       13.0         STA     DATAAP          ;SAVE ACTUAL DATA PARITY
4343  3E  01            7.0         MVI     A,$01           ;LOAD EXPECTED DATA
```

```
1476  4345  32  44  48  13.0          STA     DATAEP          ;STORE EXPECTED DATA
1477  4348  CD  49  47  18.0          CALL    CKAMT           ;
1478  434B  DA  09  43  10.0          JC      FIFOL1          ;LOOP ON ERROR
1479  434E                           ENDTST  TST01X
 (1)                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  434E                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  434E  CD  06  28  18.0                             CALL    REQST
 (2)  4351  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4352  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  4354  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4356  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4357  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4358  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  435B  3D                 4.0    DCR     A               ;DOWNCOUNT
 (1)  435C  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  435F  F2  05  43  10.0          JP      TST01X          ;DO TEST UNTIL TILL = 0
1480
1481
```

```
                                .SBTTL  TEST 02 - READ CHANNEL FIFO STATIC TEST/ALL ZEROS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*READ CHANNEL - FIFO STATIC TEST - ALL ZEROS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE FIFO'S IN PLO BYPASS MODE 3 WHERE ALL 9 FIFO'S ARE
                                ;*PARALLEL LOADED WITH ZEROS FROM THE READ CHANNEL TEST DATA BIT.  ONCE
                                ;*LOADED THE READ PATH RUNS A SPECIAL TEST ROUTINE TO READ ONE BIT FROM
                                ;*EACH FIFO AND LOAD IT IN THE DATA REGISTER.  THEN THIS REGISTER IS
                                ;*COMPARED WITH THE VALUE WRITTEN AND ANY ERRORS ARE REPORTED.  THE FAILING
                                ;*M8950 MODULE CAN BE LOCATED BY THE BIT POSITION OF THE FAILING DATA BIT(S).
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     SET PLO BYPASS MODE 3
                                ;*     CLOCK THE FIFO'S
                                ;*     ISSUE THE CLEAR ALL FIFO'S COMMAND
                                ;*     DELAY 5 MICRO SECONDS
                                ;*     STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*     CLEAR THE FIFO TEST DATA BIT
                                ;*     CLOCK THE TEST DATA INTO THE FIFO'S
                                ;*     SET THE SUBSYSTEM IN THE PE READ MODE
                                ;*     ISSUE THE FIFO READ COMMAND
                                ;*     ENABLE THE READ PATH AND READ CHANNELS CLOCK
                                ;*     DELAY 5 MICRO SECONDS
                                ;*     STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*     READ THE FIFO DATA (9 BITS)
                                ;*     SET UP THE EXPECTED DATA OF ALL ZEROS
                                ;*     CALL SUBROUTINE CKAMT
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 02
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-FIFO STATUS TEST-ALL ZEROS
                                ;*M8950, M8953
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 02
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-FIFO STATUS TEST-ALL ZEROS
```

```
1525                    ;*M8950, M8953
1526                    ;*CHANNEL 1 FAILED
1527                    ;*ACTUAL = NNNN
1528                    ;*EXPECTED = NNNN
1529                    ;*
1530                    ;*RPM2 MICRO TEST 02
1531                    ;*RPM2 MICRO ERROR 11
1532                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1533                    ;*M8950, M8953
1534                    ;*CHANNEL 2 FAILED
1535                    ;*ACTUAL = NNNN
1536                    ;*EXPECTED = NNNN
1537                    ;*
1538                    ;*RPM2 MICRO TEST 02
1539                    ;*RPM2 MICRO ERROR 12
1540                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1541                    ;*M8950, M8953
1542                    ;*CHANNEL 3 FAILED
1543                    ;*ACTUAL = NNNN
1544                    ;*EXPECTED = NNNN
1545                    ;*
1546                    ;*RPM2 MICRO TEST 02
1547                    ;*RPM2 MICRO ERROR 13
1548                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1549                    ;*M8950, M8953
1550                    ;*CHANNEL 4 FAILED
1551                    ;*ACTUAL = NNNN
1552                    ;*EXPECTED = NNNN
1553                    ;*
1554                    ;*RPM2 MICRO TEST 02
1555                    ;*RPM2 MICRO ERROR 14
1556                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1557                    ;*M8950, M8953
1558                    ;*CHANNEL 5 FAILED
1559                    ;*ACTUAL = NNNN
1560                    ;*EXPECTED = NNNN
1561                    ;*
1562                    ;*RPM2 MICRO TEST 02
1563                    ;*RPM2 MICRO ERROR 15
1564                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1565                    ;*M8950, M8953
1566                    ;*CHANNEL 6 FAILED
1567                    ;*ACTUAL = NNNN
1568                    ;*EXPECTED = NNNN
1569                    ;*
1570                    ;*RPM2 MICRO TEST 02
1571                    ;*RPM2 MICRO ERROR 16
1572                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1573                    ;*M8950, M8953
1574                    ;*CHANNEL 7 FAILED
1575                    ;*ACTUAL = NNNN
1576                    ;*EXPECTED = NNNN
1577                    ;*
1578                    ;*RPM2 MICRO TEST 02
```

```
                                        ;*RPM2 MICRO ERROR 17
                                        ;*RPM2-FIFO STATUS TEST-ALL ZEROS
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        S
                                        ; ***************************************************************************
                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
                                                MVI     A,@2            ;DEFINE THE TEST NUMBER
                                                CALL    TSET            ;SETUP THE TEST
                                        ;%RPM2-FIFO STATIC TEST-ALL ZEROS
                                        ;&M8950, M8953
                                        TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
                                                OUT     CLKCTL          ;
                                                                        ;
                                        FIFOL2: MVI     A,R.PLO1!R.PLO0 ;SET PLO BYPASS MODE 3
                                                OUT     RPCTL           ;
                                                OUT     RFIFOL          ;CLOCK THE FIFO
                                                MVI     A,RCLRT         ;LOAD ALL FIFO'S CLEAR COMMAND
                                                OUT     RCMD            ;ISSUE FIFO CLEAR
                                                NOP                     ;WAIT
                                                NOP                     ;WAIT
                                                NOP                     ;WAIT
                                                NOP                     ;WAIT
                                                MVI     A,R.PLO1!R.PLO0!R.STPC
                                                OUT     RPCTL           ;STOP THE RMC WITH FIFO DATA
                                                                        ;OF ONES
                                                OUT     RFIFOL          ;CLOCK ''0'S'' INTO THE FIFO
                                                XRA     A               ;
                                                OUT     WMCCTL          ;PUT MACHINE IN PE MODE
                                                MVI     A,FIFORD        ;LOAD THE FIFO READ COMMAND
                                                OUT     RCMD            ;
                                                                        ;
                                                MVI     A,R.PLO0!R.PLO1 ;RUN THE RMC
                                                OUT     RPCTL           ;
                                                NOP                     ;WAIT
                                                NOP                     ;WAIT
                                                NOP                     ;WAIT
                                                NOP                     ;WAIT
                                                MVI     A,R.PLO0!R.PLO1!R.STPC  ;STOP THE RMC
                                                OUT     RPCTL
                                                IN      RDATA           ;READ THE FIFO DATA
                                                STA     DATAA           ;STORE ACTUAL DATA
                                                XRA     A               ;LOAD EXPECTED DATA
                                                STA     DATAE           ;STORE EXPECTED DATA
                                                IN      RPSTA           ;GET PARITY BITS
                                                ANI     R.DATA          ;GET DATA PARITY
                                                RLC                     ;JUSTIFY THE
                                                RLC                     ;ACTUAL PARITY DATA
                                                STA     DATAAP          ;SAVE ACTUAL DATA PARITY
                                                XRA     A               ;LOAD EXPECTED DATA PARITY
```

```
1630  43A5  32  44  48  13.0          STA     DATAEP          ;STORE EXPECTED DATA PARITY
1631  43A8  CD  49  47  18.0          CALL    CKAMT           ;CHECK FOR BAD READ CHANNELS
1632  43AB  DA  6B  43  10.0          JC      FIFOL2          ;LOOP ON ERROR
1633  43AE                           ENDTST  TST02X
 (1)                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43AE                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43AE  CD  06  28  18.0                         CALL    REQST
 (2)  43B1  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  43B2  00  00                                   .WORD                   ;SYSTEM '*' COUNT
 (2)  43B4  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43B6  00                                       .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  43B7  07                                       .BYTE   7               ;REQUEST CODE
 (1)  43B8  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  43BB  3D                 4.0    DCR     A               ;DOWNCOUNT
 (1)  43BC  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  43BF  F2  67  43  10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1634
1635
```

```
                                .SBTTL  TEST 03 - READ CHANNEL FIFO STATIC TEST/ALTERNATING ZERO/ONES
1638  43C2                      ST
                                ; ******************************************************************************
                                ; TEST TITLE
                                ; ---- -----
                                ;*READ CHANNEL - FIFO STATIC TEST - ALTERNATING ZERO/ONES
1640  43C2                      SD
                                ; ******************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE FIFO'S IN PLO BYPASS MODE 3 WHERE ALL 9 FIFO'S ARE
                                ;*PARALLEL LOADED WITH ALTERNATE ZEROS/ONES FROM THE READ CHANNEL TEST
                                ;*DATA BIT.  ONCE LOADED THE READ PATH RUNS A SPECIAL TEST ROUTINE TO
                                ;*READ ONE BIT FORM EACH FIFO AND LOAD IT IN THE DATA REGISTER.  THEN
                                ;*THIS REGISTER IS COMPARED WITH THE VALUE WRITTEN AND ANY ERRORS ARE
                                ;*REPORTED.  THE FAILING M8950 MODULE CAN BE LOCATED BY THE BIT POSITION
                                ;*OF THE FAILING DATA BIT(S).
1648  43C2                      SP
                                ; ******************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    SET THE READ PASS COUNT TO 1
                                ;*    SET PLO BYPASS MODE 3
                                ;*    CLOCK THE FIFO'S
                                ;*    ISSUE CLEAR ALL FIFO'S COMMAND
                                ;*    DELAY 5 MICRO SECONDS
                                ;*    SET THE LOOP COUNTER TO 5
                                ;*    BGNDO
                                ;*       :    LOAD FIFO DATA OF ZERO
                                ;*       :    CLOCK THE DATA INTO THE FIFO
                                ;*       :    LOAD FIFO DATA OF ONE
                                ;*       :    CLOCK THE DATA INTO THE FIFO
                                ;*       :    DECREMENT THE LOOP COUNTER
                                ;*       :    DO UNTIL LOOP COUNTER=ZERO
                                ;*    ENDDO
                                ;*    BGNDO
                                ;*       :    PUT THE SUBSYSTEM IN READ PE MODE
                                ;*       :    ISSUE THE FIFO READ COMMAND
                                ;*       :    ENABLE THE READ PATH AND READ CHANNEL CLOCKS
                                ;*       :    DELAY 5 MICRO SECONDS
                                ;*       :    DISABLE THE READ PATH AND READ CHANNEL CLOCKS
                                ;*       :    READ THE FIFO DATA (9 BITS)
                                ;*       :    GET THE EXPECTED FIFO DATA FROM THE EXPECTED DATA TABLE
                                ;*       .    CALL SUBROUTINE ''CKAMT''
                                ;*       :    INCREMENT THE READ PASS COUNT
                                ;*       :    DO UNTIL THE READ PASS COUNT = 13(8)
                                ;*    ENDDO
                                ;*ENDTST
1678  43C2                      SE
                                ; ******************************************************************************
                                ;*ERRORS
                                ;*------
```

```
1679                    ;*RPM2 MICRO TEST 03
1680                    ;*RPM2 MICRO ERROR 07
1681                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1682                    ;*M8950, M8953
1683                    ;*CHANNEL 0 FAILED
1684                    ;*ACTUAL = NNNN
1685                    ;*EXPECTED - NNNN
1686                    ;*
1687                    ;*RPM2 MICRO TEST 03
1688                    ;*RPM2 MICRO ERROR 10
1689                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1690                    ;*M8950, M8953
1691                    ;*CHANNEL 1 FAILED
1692                    ;*ACTUAL - NNNN
1693                    ;*EXPECTED = NNNN
1694                    ;*
1695                    ;*RPM2 MICRO TEST 03
1696                    ;*RPM2 MICRO ERROR 11
1697                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1698                    ;*M8950, M8953
1699                    ;*CHANNEL 2 FAILED
1700                    ;*ACTUAL = NNNN
1701                    ;*EXPECTED = NNNN
1702                    ;*
1703                    ;*RPM2 MICRO TEST 03
1704                    ;*RPM2 MICRO ERROR 12
1705                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1706                    ;*M8950, M8953
1707                    ;*CHANNEL 3 FAILED
1708                    ;*ACTUAL = NNNN
1709                    ;*EXPECTED = NNNN
1710                    ;*
1711                    ;*RPM2 MICRO TEST 03
1712                    ;*RPM2 MICRO ERROR 13
1713                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1714                    ;*M8950, M8953
1715                    ;*CHANNEL 4 FAILED
1716                    ;*ACTUAL = NNNN
1717                    ;*EXPECTED = NNNN
1718                    ;*
1719                    ;*RPM2 MICRO TEST 03
1720                    ;*RPM2 MICRO ERROR 14
1721                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1722                    ;*M8950, M8953
1723                    ;*CHANNEL 5 FAILED
1724                    ;*ACTUAL = NNNN
1725                    ;*EXPECTED = NNNN
1726                    ;*
1727                    ;*RPM2 MICRO TEST 03
1728                    ;*RPM2 MICRO ERROR 15
1729                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1730                    ;*M8950, M8953
1731                    ;*CHANNEL 6 FAILED
1732                    ;*ACTUAL = NNNN
```

```
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM2 MICRO TEST 03
                                        ;*RPM2 MICRO ERROR 16
                                        ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
                                        ;*M8950, M8953
                                        ;*CHANNEL 7 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM2 MICRO TEST 03
                                        ;*RPM2 MICRO ERROR 17
                                        ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
1751  43C2                              $
 (1)                                    ; ************************************************************************************
1752
1753  43C2                              TEST3:  TESTX   @3              ;INITIALIZE THE TEST
 (1)  43C2  3E  03              7.0                     MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  43C4  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1754                                    ;%RPM2-FIFO STATIC TEST-ALTERNATE ONES/ZEROS
1755                                    ;&M8950, M8953
1756
1757  43C7  3E  10                7.0   TST03X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
1758  43C9  D3  F0               10.0           OUT     CLKCTL          ;
1759  43CB  AF                    4.0           XRA     A               ;CLEAR THE FAILING PASS NUMBER
1760  43CC  32  49  48           13.0           STA     PASFL
1761  43CF  3E  01                7.0   1$:     MVI     A,@1            ;LOAD THE BEGINNING PASS COUNTER
1762  43D1  32  48  48           13.0           STA     PASCT           ;SAVE PASS COUNTER
1763  43D4  3E  30                7.0           MVI     A,R.PLO1.R.PLO0 ;SET PLO BYPASS MODE 3
1764  43D6  D3  09               10.0           OUT     RPCTL           ;
1765  43D8  D3  08               10.0           OUT     RFIFOL          ;CLOCK THE FIFO
1766  43DA  3E  0D                7.0           MVI     A,RCLRT         ;LOAD ALL FIFO'S CLEAR COMMAND
1767  43DC  D3  0B               10.0           OUT     RCMD            ;ISSUE FIFO CLEAR
1768  43DE  00                    4.0           NOP                     ;WAIT
1769  43DF  00                    4.0           NOP                     ;WAIT
1770  43E0  00                    4.0           NOP                     ;WAIT
1771  43E1  00                    4.0           NOP                     ;WAIT
1772  43E2  06  05                7.0           MVI     B,@5            ;LOAD CONTROL COUNT
1773  43E4  3E  31                7.0   2$:     MVI     A,R.PLO0.R.PLO1!R.STPC  ;LOAD FIFO DATA OF ZERO
1774  43E6  D3  09               10.0           OUT     RPCTL
1775  43E8  D3  08               10.0           OUT     RFIFOL          ;CLOCK-ZEROS INTO THE FIFOS
1776  43EA  3E  71                7.0           MVI     A,R.PLO0!R.PLO1!R.STPC!@100  ;LOAD FIFO DATA OF ONE
1777  43EC  D3  09               10.0           OUT     RPCTL
1778  43EE  D3  08               10.0           OUT     RFIFOL          ;CLOCK-ONES INTO THE FIFOS
1779  43F0  05                    4.0           DCR     B
1780  43F1  C2  E4  43           10.0           JNZ     2$
1781
1782  43F4  21  2E  48           10.0           LXI     H,FIFOEX        ;LOAD START OF EXPECTED DATA TBL.
1783  43F7  3A  48  48           13.0   3$:     LDA     PASCT           ;WRITE THE PASS COUNT TO CAS REGISTER 5
```

```
1784  43FA                                  10.0  ROUT    ROSL
 (1)  43FA  D3  8A                          10.0          OUT     ROSL            ;WRITE AC INTO ROSL
 (1)  43FC  7F                               4.0          MOV     A,A             ;RETRY LINK
1785  43FD  AF                               4.0  XRA     A                ;CLEAR CAS REGISTER 5 HIGH
1786  43FE                                       ROUT    ROSH
 (1)  43FE  D3  8B                          10.0          OUT     ROSH            ;WRITE AC INTO ROSH
 (1)  4400  7F                               4.0          MOV     A,A             ;RETRY LINK
1787  4401  AF                               4.0  XRA     A                ;
1788  4402  D3  D3                          10.0  OUT     WMCCTL           ;PUT MACHINE IN PE MODF
1789  4404  3E  6A                          7.0   MVI     A,'IFORD         ;LOAD THE FIFO READ COMMAND
1790  4406  D3  08                          10.0  OUT     RCMD             ;
1791  4408  3E  30                          7.0   MVI     A,R.PL00.R.PL01  ;RUN THE RMC
1792  440A  D3  09                          10.0  OUT     RPCTL            ;
1793  440C  00                               4.0  NOP                      ;WAIT
1794  440D  00                               4.0  NOP                      ;WAIT
1795  440E  00                               4.0  NOP                      ;WAIT
1796  440F  00                               4.0  NOP                      ;WAIT
1797  4410  3E  31                          7.0   MVI     A,R.PL00!R.PL01!R.STPC  ;STOP THE RMC
1798  4412  D3  09                          10.0  OUT     RPCTL
1799  4414  DB  17                          10.0  IN      RDATA            ;READ THE FIFO DATA
1800  4416  32  45  48                     13.0  STA     DATAA            ;STORE ACTUAL DATA
1801  4419  7E                              7.0   MOV     A,M              ;GET EXPECTED DATA
1802  441A  32  43  48                     13.0  STA     DATAE            ;STORE EXPECTED DATA
1803  441D  23                              6.0   INX     H                ;UPDATE TABLE POINTER
1804  441E  DB  15                          10.0  IN      RPSTA            ;GET PARITY BITS
1805  4420  E6  40                          7.0   ANI     R.DATA           ;GET DATA PARITY
1806  4422  07                              4.0   RLC                      ;JUSTIFY PARITY BIT
1807  4423  07                              4.0   RLC                      ;
1808  4424  32  46  48                     13.0  STA     DATAAP           ;SAVE ACTUAL DATA PARITY
1809  4427  7E                              7.0   MOV     A,M              ;GET EXPECTED DATA PARITY
1810  4428  32  44  48                     13.0  STA     DATAEP           ;SAVE EXPECTED DATA PARITY
1811  442B  E5                              12.0  PUSH    H                ;SAVE H
1812  442C  CD  49  47                     18.0  CALL    CKAMT            ;CHECK FOR BAD
1813  442F  E1                              10.0  POP     H                ;RESTORE H
1814  4430  D2  4E  44                     10.0  JNC     4$               ;NO ERROR CONTINUE
1815  4433  3A  49  48                     13.0  LDA     PASFL            ;GET THE FAILING PASS NUMBER
1816  4436  A7                              4.0   ANA     A                ;SET THE CONDITION BITS
1817  4437  CA  45  44                     10.0  JZ      5$               ;FIRST ERROR GO INITIALIZE THE LOOP
1818  443A  47                              4.0   MOV     B,A              ;SAVE THE FAILING PASS NUMBER
1819  443B  3A  48  48                     13.0  LDA     PASCT            ;GET THE NUMBER OF THIS PASS
1820  443E  B8                              4.0   CMP     B                ;EQUAL?
1821  443F  C2  4E  44                     10.0  JNZ     4$               ;NO - CONTINUE WITH THE TEST
1822  4442  C3  CF  43                     10.0  JMP     1$               ;YES - RESTART THE TEST
1823  4445  3A  48  48                     13.0  5$:     LDA     PASCT    ;GET THE PASS COUNT
1824  4448  32  49  48                     13.0          STA     PASFL    ;STORE AS THE FAILING PASS NUMBER
1825  444B  C3  CF  43                     10.0  JMP     1$               ;RESTART THE TEST
1826  444E  23                              6.0   4$:     INX     H        ;UPDATE POINTER
1827  444F  3A  48  48                     13.0  LDA     PASCT            ;GET THE PASS COUNT
1828  4452  3C                              4.0   INR     A                ;UPDATE THE COUNTER
1829  4453  32  48  48                     13.0  STA     PASCT            ;RESTORE TO MEMORY
1830  4456  01  42  48                     10.0  LXI     B,FIFOEN         ;GET END OF TABLE
1831  4459  79                              4.0   MOV     A,C
1832  445A  BD                              4.0   CMP     L                ;DONE?
1833  445B  C2  F7  43                     10.0  JNZ     3$               ;NO
```

```
1834  445E                                    ENDTST  TST03X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  445E                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  445E   CD   06   28        18.0                                 CALL    REQST
 (2)  4461   00                                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  4462   00   00                                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  4464   00   00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4466   00                                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4467   07                                                       .BYTE   7               ;REQUEST CODE
 (1)  4468   3A   9A   4F       13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  446B   3D                 4.0                 DCR     A               ;DOWNCOUNT
 (1)  446C   32   9A   4F       13.0                STA     ITERA           ;SAVE COUNT
 (1)  446F   F2   67   43       10.0                JP      TST03X          ;DO TEST UNTIL TILL = 0
1835
1836
```

```
                                .SBTTL  TEST 04 - READ DATA PATH TEST - FROM TU PORT #0
1839  4472                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1841  4472                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 0 TO THE READ
                                ;*CHANNELS.
1844  4472                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 SELECTED BY USER
                                ;*        THEN-CONTINUE
                                ;*        ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #0 SELECT CODE
                                ;*    CALL SUBROUTINE RDSET
                                ;*ENDTST
1854  4472                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 04
                                ;*RPM2 MICRO ERROR 01
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED!
                                ;*
                                ;*RPM2 MICRO TEST 04
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 04
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 04
                                ;*RPM2 MICRO ERROR 11
```

```
1880                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1881                    ;*M8950, M8953, M8955'S
1882                    ;*CHANNEL 2 FAILED
1883                    ;*ACTUAL = NNNN
1884                    ;*EXPECTED = NNNN
1885                    ;*
1886                    ;*RPM2 MICRO TEST 04
1887                    ;*RPM2 MICRO ERROR 12
1888                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1889                    ;*M8950, M8953, M8955'S
1890                    ;*CHANNEL 3 FAILED
1891                    ;*ACTUAL = NNNN
1892                    ;*EXPECTED = NNNN
1893                    ;*
1894                    ;*RPM2 MICRO TEST 04
1895                    ;*RPM2 MICRO ERROR 13
1896                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1897                    ;*M8950, M8953, M8955'S
1898                    ;*CHANNEL 4 FAILED
1899                    ;*ACTUAL = NNNN
1900                    ;*EXPECTED = NNNN
1901                    ;*
1902                    ;*RPM2 MICRO TEST 04
1903                    ;*RPM2 MICRO ERROR 14
1904                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1905                    ;*M8950, M8953, M8955'S
1906                    ;*CHANNEL 5 FAILED
1907                    ;*ACTUAL = NNNN
1908                    ;*EXPECTED = NNNN
1909                    ;*
1910                    ;*RPM2 MICRO TEST 04
1911                    ;*RPM2 MICRO ERROR 15
1912                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1913                    ;*M8950, M8953, M8955'S
1914                    ;*CHANNEL 6 FAILED
1915                    ;*ACTUAL = NNNN
1916                    ;*EXPECTED = NNNN
1917                    ;*
1918                    ;*RPM2 MICRO TEST 04
1919                    ;*RPM2 MICRO ERROR 16
1920                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1921                    ;*M8950, M8953, M8955'S
1922                    ;*CHANNEL 7 FAILED
1923                    ;*ACTUAL = NNNN
1924                    ;*EXPECTED = NNNN
1925                    ;*
1926                    ;*RPM2 MICRO TEST 04
1927                    ;*RPM2 MICRO ERROR 17
1928                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1929                    ;*M8950, M8953, M8955'S
1930                    ;*CHANNEL P FAILED
1931                    ;*ACTUAL = NNNN
1932                    ;*EXPECTED = NNNN
1933                    ;*BYTE/SCLK COUNT NUMBER   LLL
```

```
1934  4472                                          S
 (1)                                                ; ***************************************************************************
1935
1936  4472                                          TEST4:  TESTX   @004            ;INITIALIZE THE TEST
 (1)  4472  3E  04                        7.0                MVI     A,@004          ;DEFINE THE TEST NUMBER
 (1)  4474  CD  03  28                   18.0                CALL    TSET            ;SETUP THE TEST
1937                                                ;%RPM2-READ DATA PATH TEST-FROM TU PORT #0
1938                                                ;&M8950, M8953, M8955'S
1939
1940  4477                                                  REQ     @7,0,0,0,0
 (1)  4477  CD  06  28                   18.0                CALL    REQST
 (1)  447A  00                                               .BYTE   0               ;DATA PATTERN NUMBER
 (1)  447B  00  00                                           .WORD   0               ;SYSTEM '0' COUNT
 (1)  447D  00  00                                           .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  447F  00                                               .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  4480  07                                               .BYTE   @7              ;REQUEST CODE
1941
1942  4481                                                  RIN     R12L            ;GET THE UNITS DESIRED
 (1)  4481  DB  94                       10.0                IN      R12L                    ;READ R12L INTO AC
 (1)  4483  7F                            4.0                MOV     A,A                     ;RETRY LINK
1943  4484  32  47  48                   13.0                STA     UNITMP          ;STORE IN MEMORY
1944  4487  A7                            4.0                ANA     A               ;SET THE CONDITION CODE
1945  4488  C2  99  44                   10.0                JNZ     TST04X          ;GO TEST SPECIFIED UNITS
1946  448B                                                  ERR     EXIT,DUMMY
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  448B  CD  09  28                   18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448E  01                                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  448F  00                                               .BYTE
 (1)  4490  CD  15  28                   18.0        DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4493  DA  2C  46                   10.0                JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1947                                                ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1948                                                ;>FATAL ERROR - TEST ABORTED!
1949  4496  C3  2C  46                   10.0                JMP     EXIT
1950  4499  3A  47  48                   13.0        TST04X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
1951  449C  E6  01                        7.0                ANI     $01             ;TEST FOR TU PORT #0
1952  449E  CA  BD  44                   10.0                JZ      TEST5           ;NO-GO TO NEXT TEST
1953  44A1  CD  1E  47                   18.0                CALL    CLEAR           ;
1954  44A4  3E  00                        7.0                MVI     A,0             ;
1955
1956  44A6  CD  98  46                   18.0                CALL    RDSET           ;GO SET UP THE READ CHANNELS
1957  44A9                                                  ENDTST  TST04X
 (1)                                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44A9                                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44A9  CD  06  28                   18.0                                CALL    REQST
 (2)  44AC  00                                                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  44AD  00  00                                                           .WORD                   ;SYSTEM '-' COUNT
 (2)  44AF  00  00                                                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44B1  00                                                               .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  44B2  07                                                               .BYTE   7               ;REQUEST CODE
 (1)  44B3  3A  9A  4F                   13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  44B6  3D                            4.0                DCR     A               ;DOWNCOUNT
 (1)  44B7  32  9A  4F                   13.0                STA     ITERA           ;SAVE COUNT
 (1)  44BA  F2  99  44                   10.0                JP      TST04X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 05 - READ DATA PATH TEST - FROM TU PORT #1
44BD                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
44BD                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 1 TO THE READ
                                ;*CHANNELS.
44BD                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 SELECTED BY USER
                                ;*        THEN-CONTINUE
                                ;*        ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #1 SELECT CODE
                                ;*    CALL SUBROUTINE RDSET
                                ;*ENDTST
44BD                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 05
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 05
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 05
                                ;*RPM2 MICRO ERROR 11
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 05
```

```
2001                                    ;*RPM2 MICRO ERROR 12
2002                                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2003                                    ;*M8950, M8953, M8955'S
2004                                    ;*CHANNEL 3 FAILED
2005                                    ;*ACTUAL = NNNN
2006                                    ;*EXPECTED = NNNN
2007                                    ;*
2008                                    ;*RPM2 MICRO TEST 05
2009                                    ;*RPM2 MICRO ERROR 13
2010                                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2011                                    ;*M8950, M8953, M8955'S
2012                                    ;*CHANNEL 4 FAILED
2013                                    ;*ACTUAL = NNNN
2014                                    ;*EXPECTED = NNNN
2015                                    ;*
2016                                    ;*RPM2 MICRO TEST 05
2017                                    ;*RPM2 MICRO ERROR 14
2018                                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2019                                    ;*M8950, M8953, M8955'S
2020                                    ;*CHANNEL 5 FAILED
2021                                    ;*ACTUAL = NNNN
2022                                    ;*EXPECTED = NNNN
2023                                    ;*
2024                                    ;*RPM2 MICRO TEST 05
2025                                    ;*RPM2 MICRO ERROR 15
2026                                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2027                                    ;*M8950, M8953, M8955'S
2028                                    ;*CHANNEL 6 FAILED
2029                                    ;*ACTUAL = NNNN
2030                                    ;*EXPECTED = NNNN
2031                                    ;*
2032                                    ;*RPM2 MICRO TEST 05
2033                                    ;*RPM2 MICRO ERROR 16
2034                                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2035                                    ;*M8950, M8953, M8955'S
2036                                    ;*CHANNEL 7 FAILED
2037                                    ;*ACTUAL = NNNN
2038                                    ;*EXPECTED = NNNN
2039                                    ;*
2040                                    ;*RPM2 MICRO TEST 05
2041                                    ;*RPM2 MICRO ERROR 17
2042                                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2043                                    ;*M8950, M8953, M8955'S
2044                                    ;*CHANNEL P FAILED
2045                                    ;*ACTUAL = NNNN
2046                                    ;*EXPECTED = NNNN
2047                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2048  44BD                              S
 (1)                                    ; ***************************************************************************************
```

```
2050  44BD                                  TEST5:  TESTX   @005            ;INITIALIZE THE TEST
 (1)  44BD  3E  05                    7.0                    MVI     A,@005          ;DEFINE THE TEST NUMBER
 (1)  44BF  CD  03  28               18.0                    CALL    TSET            ;SETUP THE TEST
2051                                         ;%RPM2-READ DATA PATH TEST-FROM TU PORT #1
2052                                         ;&M8950, M8953, M8955'S
2053  44C2  3A  47  48                13.0  TST05X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2054  44C5  E6  02                    7.0          ANI     $02             ;TEST FOR TU PORT #1
2055  44C7  CA  E6  44                10.0          JZ      TEST6           ;NO-GO TO NEXT TEST
2056  44CA  CD  1E  47                18.0          CALL    CLEAR           ;
2057  44CD  3E  01                    7.0          MVI     A,@1
2058  44CF  CD  9B  46                18.0          CALL    RDSET           ;GO SET UP THE READ CHANNELS
2059  44D2                                        ENDTST  TST05X
 (1)                                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44D2                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44D2  CD  06  28               18.0                    CALL    REQST
 (2)  44D5  00                                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  44D6  00  00                                           .WORD                   ;SYSTEM "" COUNT
 (2)  44D8  00  00                                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SK:P
 (2)  44DA  00                                               .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44DB  07                                               .BYTE   7               ;REQUEST CODE
 (1)  44DC  3A  9A  4F               13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  44DF  3D                        4.0          DCR     A               ;DOWNCOUNT
 (1)  44E0  32  9A  4F               13.0          STA     ITERA           ;SAVE COUNT
 (1)  44E3  F2  C2  44               10.0          JP      TST05X          ;DO TEST UNTIL TILL = 0
2060
2061
```

```
                                .SBTTL  TEST 06 - READ DATA PATH TEST - FROM TU PORT #2
44E6                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
44F6                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 2 TO THE READ
                                ;*CHANNELS.
44E6                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 SELECTED BY USER
                                ;*        THEN-CONTINUE
                                ;*        ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #2 SELECT CODE
                                ;*    CALL SUBROUTINE RDSET
                                ;*ENDTST
44E6                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 06
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 06
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 06
                                ;*RPM2 MICRO ERROR 11
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 06
```

```
2105                                ;*RPM2 MICRO ERROR 12
2106                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2107                                ;*M8950, M8953, M8955'S
2108                                ;*CHANNEL 3 FAILED
2109                                ;*ACTUAL = NNNN
2110                                ;*EXPECTED = NNNN
2111                                ;*
2112                                ;*RPM2 MICRO TEST 06
2113                                ;*RPM2 MICRO ERROR 13
2114                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2115                                ;*M8950, M8953, M8955'S
2116                                ;*CHANNEL 4 FAILED
2117                                ;*ACTUAL = NNNN
2118                                ;*EXPECTED = NNNN
2119                                ;*
2120                                ;*RPM2 MICRO TEST 06
2121                                ;*RPM2 MICRO ERROR 14
2122                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2123                                ;*M8950, M8953, M8955'S
2124                                ;*CHANNEL 5 FAILED
2125                                ;*ACTUAL = NNNN
2126                                ;*EXPECTED = NNNN
2127                                ;*
2128                                ;*RPM2 MICRO TEST 06
2129                                ;*RPM2 MICRO ERROR 15
2130                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2131                                ;*M8950, M8953, M8955'S
2132                                ;*CHANNEL 6 FAILED
2133                                ;*ACTUAL = NNNN
2134                                ;*EXPECTED = NNNN
2135                                ;*
2136                                ;*RPM2 MICRO TEST 06
2137                                ;*RPM2 MICRO ERROR 16
2138                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2139                                ;*M8950, M8953, M8955'S
2140                                ;*CHANNEL 7 FAILED
2141                                ;*ACTUAL = NNNN
2142                                ;*EXPECTED = NNNN
2143                                ;*
2144                                ;*RPM2 MICRO TEST 06
2145                                ;*RPM2 MICRO ERROR 17
2146                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2147                                ;*M8950, M8953, M8955'S
2148                                ;*CHANNEL P FAILED
2149                                ;*ACTUAL = NNNN
2150                                ;*EXPECTED = NNNN
2151                                ;*BYTE/SCLK COUNT NUMBER - LLL
2152  44E6                          S
 (1)                                ; ***************************************************************************************
```

```
2154  44E6                                  TEST6:  TESTX   @006            ;INITIALIZE THE TEST
 (1)  44E6  3E  06              7.0                    MVI     A,@006          ;DEFINE THE TEST NUMBER
 (1)  44E8  CD  03  28         18.0                    CALL    TSET            ;SETUP THE TEST
2155                                  ;%RPM2-READ DATA PATH TEST-FROM TU PORT #2
2156                                  ;&M8950, M8953, M8955'S
2157
2158  44EB  3A  47  48         13.0  TST06X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2159  44EE  E6  04                7.0          ANI     $04             ;TEST FOR TU PORT #2
2160  44F0  CA  0F  45           10.0          JZ      TEST7           ;NO-GO TO NEXT TEST
2161  44F3  CD  1E  47           18.0          CALL    CLEAR
2162  44F6  3E  02                7.0          MVI     A,@2
2163  44F8  CD  9B  46           18.0          CALL    RDSET           ;GO SET UP THE READ CHANNELS
2164  44FB                                    ENDTST  TST06X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44FB                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44FB  CD  06  28         18.0                            CALL    REQST
 (2)  44FE  00                                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  44FF  00  00                                             .WORD                   ;SYSTEM '*' COUNT
 (2)  4501  00  00                                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4503  00                                                 .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4504  07                                                 .BYTE   7               ;REQUEST CODE
 (1)  4505  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4508  3D                  4.0          DCR     A               ;DOWNCOUNT
 (1)  4509  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  450C  F2  EB  44        -10.0          JP      TST06X          ;DO TEST UNTIL TILL = 0
2165
2166
```

```
                                .SBTTL  TEST 07 - READ DATA PATH TEST - FROM TU PORT #3
450F                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*----- -----
                                ;*RPM2 READ DATA PATH TEST-FROM TU PORT #3
450F                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 3 TO THE READ
                                ;*CHANNELS.
450F                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 SELECTED BY USER
                                ;*        THEN-CONTINUE
                                ;*        ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #3 SELECT CODE
                                ;*    CALL SUBROUTINE RDSET
                                ;*ENDTST
450F                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 07
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED - NNNN
                                ;*
                                ;*RPM2 MICRO TEST 07
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 07
                                ;*RPM2 MICRO ERROR 11
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 07
```

```
2210                    ;*RPM2 MICRO ERROR 12
2211                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2212                    ;*M8950, M8953, M8955'S
2213                    ;*CHANNEL 3 FAILED
2214                    ;*ACTUAL = NNNN
2215                    ;*EXPECTED = NNNN
2216                    ;*
2217                    ;*RPM2 MICRO TEST 07
2218                    ;*RPM2 MICRO ERROR 13
2219                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2220                    ;*M8950, M8953, M8955'S
2221                    ;*CHANNEL 4 FAILED
2222                    ;*ACTUAL = NNNN
2223                    ;*EXPECTED = NNNN
2224                    ;*
2225                    ;*RPM2 MICRO TEST 07
2226                    ;*RPM2 MICRO ERROR 14
2227                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2228                    ;*M8950, M8953, M8955'S
2229                    ;*CHANNEL 5 FAILED
2230                    ;*ACTUAL = NNNN
2231                    ;*EXPECTED = NNNN
2232                    ;*
2233                    ;*RPM2 MICRO TEST 07
2234                    ;*RPM2 MICRO ERROR 15
2235                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2236                    ;*M8950, M8953, M8955'S
2237                    ;*CHANNEL 6 FAILED
2238                    ;*ACTUAL = NNNN
2239                    ;*EXPECTED = NNNN
2240                    ;*
2241                    ;*RPM2 MICRO TEST 07
2242                    ;*RPM2 MICRO ERROR 16
2243                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2244                    ;*M8950, M8953, M8955'S
2245                    ;*CHANNEL 7 FAILED
2246                    ;*ACTUAL - NNNN
2247                    ;*EXPECTED = NNNN
2248                    ;*
2249                    ;*RPM2 MICRO TEST 07
2250                    ;*RPM2 MICRO ERROR 17
2251                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2252                    ;*M8950, M8953, M8955'S
2253                    ;*CHANNEL P FAILED
2254                    ;*ACTUAL = NNNN
2255                    ;*EXPECTED = NNNN
2256                    ;*BYTE/SCLK COUNT NUMBER   LLL
2257  450F             S
 (1)                    ; ***************************************************************************
2258
```

```
2260  450F                                  TEST7:  TESTX   @007             ;INITIALIZE THE TEST
 (1)  450F  3E  07              7.0                      MVI     A,@007          ;DEFINE THE TEST NUMBER
 (1)  4511  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
2261                                   ;%RPM2-READ DATA PATH TEST-FROM TU PORT #3
2262                                   ;&M8950, M8953, M8955'S
2263  4514  3A  47  48         13.0    TST07X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2264  4517  E6  10                7.0          ANI     $010            ;TEST FOR TU PORT #3
2265  4519  CA  38  45          10.0          JZ      TEST10          ;NO-EXIT TEST
2266  451C  CD  1E  47          18.0          CALL    CLEAR
2267  451F  3E  03                7.0          MVI     A,@3
2268  4521  CD  9B  46          18.0          CALL    RDSET           ;GO SET UP THE READ CHANNELS
2269  4524                                    ENDTST  TST07X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4524                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4524  CD  06  28         18.0                      CALL    REQST
 (2)  4527  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4528  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  452A  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  452C  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  452D  07                                           .BYTE   7               ;REQUEST CODE
 (1)  452E  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4531  3D                 4.0           DCR     A               ;DOWNCOUNT
 (1)  4532  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  4535  F2  14  45         10.0          JP      TST07X          ;DO TEST UNTIL TILL = 0
2270
2271
```

```
                                .SBTTL  TEST 10 - VELOCITY BIT JAM TEST

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*VELOCITY BIT JAM TEST
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST ASSURES THAT THE VELOCITY OK BIT CAN BE SET BY WRITING THE
                                ;*VELOCITY OK JAM BIT IN THE READ PATH CONTROL WORD.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      SET NORMAL READ PATH CLOCK
                                ;*      STOP THE READ PATH AND READ CHANNELS
                                ;*      SET THE JAM VELOCITY OK BIT
                                ;*      INPUT THE VELOCITY OK STATUS BIT
                                ;*        IF VELOCITY OK STATUS BIT-1
                                ;*            THEN-CONTINUE
                                ;*            ELSE-ERROR
                                ;*        ENDIF
                                ;*      CLEAR THE JAM VELOCITY OK BIT
                                ;*      DELAY 50 MICRO SECONDS
                                ;*      INPUT THE VELOCITY OK STATUS BIT
                                ;*        IF VELOCITY OK STATUS BIT 0
                                ;*            THEN-CONTINUE
                                ;*            ELSE-ERROR
                                ;*        ENDIF
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 10
                                ;*RPM2 MICRO ERROR 02
                                ;*RPM2-VELOCITY LOGIC TEST
                                ;*M8953
                                ;*VELOCITY 'OK'' BIT NOT SET WHEN JAMMED SET
                                ;*
                                ;*RPM2 MICRO TEST 10
                                ;*RPM2 MICRO ERROR 03
                                ;*RPM2-VELOCITY LOGIC TEST
                                ;*M8953
                                ;*VELOCITY 'OK'' BIT STILL SET WHEN JAMMED RESET
                                S
                                ; ************************************************************************
```

```
                                        TEST10: TESTX   @10              ;INITIALIZE THE TEST
2312 4538                                               MVI     A,@10           ;DEFINE THE TEST NUMBER
 (1) 4538 3E 08              7.0                        CALL    TSET            ;SETUP THE TEST
 (1) 453A CD 03 28          18.0
                                        ;%RPM2-VELOCITY LOGIC TEST
2313
                                        ;@M8953
2314
2315 453C 3E 10             7.0         TST10X: MVI     A,RDCLK         ;SET THE NORMAL READ PATH CLOCK
2316 453E D3 F0            10.0                 OUT     CLKCTL          ;
2317 4541 3E 05             7.0         VOKLP0: MVI     A,R.STPC!R.JVOK ;STOP THE READ MACHINE AND SET
2318 4543 D3 09            10.0                 OUT     RPCTL           ;THE VELOCITY OK BIT.
2319 4545 00                4.0                 NOP
2320 4546 DB 01            10.0                 IN      RPCHI           ;GET THE VELOCITY OK BIT
2321 4548 E6 80             7.0                 ANI     R.VOK           ;REMOVE UNWANTED BITS
2322 454A C2 52 45         10.0                 JNZ     VOKCN0
2323 454D                                       ERR     VOKLP0,VOKCN0
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 454D CD 09 28          18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      0002                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4550 02                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4551 00                                             .BYTE
 (1) 4552 CD 15 28          18.0                 VOKCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4555 DA 41 45          10.0                         JC      VOKLP0          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>VELOCITY "OK" BIT NOT SET WHEN JAMMED SET
2324
2325
2326 4558 3E 01             7.0         VOKLP1: MVI     A,R.STPC                ;RESET THE VELOCITY OK
2327 455A D3 09            10.0                 OUT     RPCTL           ;JAM BIT
2328 455C 3E 0A             7.0                 MVI     A,10            ;LOAD A 50 MICRO SECOND TIMEOUT
2329 455E 3D                4.0         1$:     DCR     A               ;DECREMENT THE COUNTER
2330 455F C2 5E 45         10.0                 JNZ     1$              ;CONTINUE UNTIL DONE
2331 4562 DB 01            10.0                 IN      RPCHI           ;GET THE VELOCITY OK CHECK BYTE
2332 4564 E6 80             7.0                 ANI     R.VOK           ;VELOCITY OK?
2333 4566 CA 6E 45         10.0                 JZ      VOKCN1          ;YES - CONTINUE
2334 4569                                       ERR     VOKLP1,VOKCN1
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 4569 CD 09 28          18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      0003                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 456C 03                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 456D 00                                             .BYTE
 (1) 456E CD 15 28          18.0                 VOKCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4571 DA 58 45          10.0                         JC      VOKLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
2335                                    ;>VELOCITY "OK" BIT STILL SET WHEN JAMMED RESET
2336 4574                                       ENDTST  TST10X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 4574                                       REQ     7                       ;MAKE CALL TO KEEP TEST ALIVE
 (2) 4574 CD 06 28          18.0                         CALL    REQST
 (2) 4577 00                                             .BYTE                   ;DATA PATTERN NUMBER
 (2) 4578 00 00                                          .WORD                   ;SYSTEM "#" COUNT
 (2) 457A 00 00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 457C 00                                             .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 457D 07                                             .BYTE   7               ;REQUEST CODE
 (1) 457E 3A 9A 4F          13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1) 4581 3D                 4.0                 DCR     A               ;DOWNCOUNT
 (1) 4582 32 9A 4F          13.0                 STA     ITERA           ;SAVE COUNT
 (1) 4585 F2 3D 45          10.0                 JP      TST10X          ;DO TEST UNTIL TILL = 0
2337
```

```
                                .SBTTL  TEST 11 - VELOCITY TEST - FROM TU PORT #0
4588                            ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*VELOCITY TEST - FROM TU PORT #0
4588                            SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                                ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                                ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 0.
4588                            SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 SELECTED BY USER
                                ;*      THEN-CONTINUE
                                ;*      ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD THE PORT #0 CONTROL ADDRESS
                                ;*    CALL SUBROUTINE VELCTL
                                ;*ENDTST
4588                            SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 11
                                ;*RPM2 MICRO ERROR 04
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #0
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT -25%
                                ;*
                                ;*RPM2 MICRO TEST 11
                                ;*RPM2 MICRO ERROR 05
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #0
                                ;*M8953, M8955'S
                                ;*VELOCITY OK NOT SET AT NOMINAL
                                ;*
                                ;*RPM2 MICRO TEST 11
                                ;*RPM2 MICRO ERROR 06
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #0
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT +25%
4588                            S
                                ; ************************************************************************
```

```
2376  4588                                    TEST11: TESTX   @11              ;INITIALIZE THE TEST
 (1)  4588  3E  09              7.0                      MVI     A,@11           ;DEFINE THE TEST NUMBER
 (1)  458A  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
2377                                    ;%RPM2-VELOCITY TEST-FROM TU PORT #0
2378                                    ;&M8953, M8955'S
2379  458D  3A  47  48         13.0    TST11X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2380  4590  E6  01                7.0            ANI     $01             ;TEST TU PORT #0
2381  4592  CA  B1  45         10.0            JZ      TEST12          ;NO-GO TO NEXT TEST
2382  4595  CD  1E  47         18.0            CALL    CLEAR           ;CLEAR ALL TU PORTS
2383  4598  3E  00                7.0            MVI     A,@0
2384  459A  CD  50  46         18.0            CALL    VELCTL
2385  459D                                    ENDTST  TST11X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  459D                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  459D  CD  06  28         18.0                      CALL    REQST
 (2)  45A0  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  45A1  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  45A3  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45A5  00                                           .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  45A6  07                                           .BYTE   7               ;REQUEST CODE
 (1)  45A7  3A  9A  4F         13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  45AA  3D                 4.0             DCR     A               ;DOWNCOUNT
 (1)  45AB  32  9A  4F         13.0            STA     ITERA           ;SAVE COUNT
 (1)  45AF  F2  8D  45         10.0            JP      TST11X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - VELOCITY TEST - FROM TU PORT #1
                               ST
                               ; ************************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               SD
                               ; ************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                               ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                               ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 1.
                               SP
                               ; ************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*     IF TU PORT #1 SELECTED BY USER
                               ;*       THEN-CONTINUE
                               ;*       ELSE-NEXT TEST
                               ;*     ENDIF
                               ;*     CALL SUBROUTINE CLEAR
                               ;*     LOAD THE PORT #1 CONTROL ADDRESS
                               ;*     CALL SUBROUTINE VELCTL
                               ;*ENDTST
                               SE
                               ; ************************************************************************
                               ;*ERRORS
                               ;*------
                               ;*RPM2 MICRO TEST 12
                               ;*RPM2 MICRO ERROR 04
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               ;*M8953, M8955'S
                               ;*VELOCITY OK SET AT -25%
                               ;*
                               ;*RPM2 MICRO TEST 12
                               ;*RPM2 MICRO ERROR 05
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               ;*M8953, M8955'S
                               ;*VELOCITY OK NOT SET AT NOMINAL
                               ;*
                               ;*RPM2 MICRO TEST 12
                               ;*RPM2 MICRO ERROR 06
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               ;*M8953, M8955'S
                               ;*VELOCITY OK SET AT +25%
                               S
                               ; ************************************************************************
```

```
2424  45B1                                TEST12: TESTX   @12              ;INITIALIZE THE TEST
 (1)  45B1  3E  0A           7.0                        MVI     A,@12       ;DEFINE THE TEST NUMBER
 (1)  45B3  CD  03  28      18.0                        CALL    TSET        ;SETUP THE TEST
2425                                ;%RPM2-VELOCITY TEST-FROM TU PORT #1
2426                                ;&M8953, M8955'S
2427  45B6  3A  47  48          13.0 TST12X: LDA     UNITMP              ;GET THE UNIT MAP BYTE
2428  45B9  E6  02                7.0          ANI     $02                 ;TEST TU PORT #1
2429  45BB  CA  DA  45           10.0          JZ      TEST13              ;NO-GO TO NEXT TEST
2430  45BE  CD  1E  47           18.0          CALL    CLEAR               ;CLEAR ALL TU PORTS
2431  45C1  3F  01                7.0          MVI     A,@1
2432  45C3  CD  50  46           18.0          CALL    VELCTL
2433  45C6                                     ENDTST  TST12X
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45C6                                     REQ     7                   ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45C6  CD  06  28      18.0                        CALL    REQST
 (2)  45C9  00                                          .BYTE               ;DATA PATTERN NUMBER
 (2)  45CA  00  00                                      .WORD               ;SYSTEM '*' COUNT
 (2)  45CC  00  00                                      .WORD               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45CE  00                                          .BYTE               ;DATA COMPARE FLAG IF -1
 (2)  45CF  07                                          .BYTE   7           ;REQUEST CODE
 (1)  45D0  3A  9A  4F      13.0          LDA     ITERA               ;GET ITERATION COUNT
 (1)  45D3  3D               4.0          DCR     A                   ;DOwNCOUNT
 (1)  45D4  32  9A  4F      13.0          STA     ITERA               ;SAVE COUNT
 (1)  45D7  F2  B6  45      10.0          JP      TST12X              ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - VELOCITY TEST - FROM TU PORT #2
45DA                            ST
                                ; **********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
45DA                            SD
                                ; **********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                                ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                                ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 2.
45DA                            SP
                                ; **********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #2 SELECTED BY USER
                                ;*       THEN-CONTINUE
                                ;*       ELSE-NEXT TEST
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     LOAD THE PORT #2 CONTROL ADDRESS
                                ;*     CALL SUBROUTINE VELCTL
                                ;*ENDTST
45DA                            SE
                                ; **********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 13
                                ;*RPM2 MICRO ERROR 04
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT -25%
                                ;*
                                ;*RPM2 MICRO TEST 13
                                ;*RPM2 MICRO ERROR 05
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
                                ;*M8953, M8955'S
                                ;*VELOCITY OK NOT SET AT NOMINAL
                                ;*
                                ;*RPM2 MICRO TEST 13
                                ;*RPM2 MICRO ERROR 06
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT +25%
45DA                            S
                                ; **********************************************************************
```

```
2472  45DA                                  TEST13: TESTX   @13              ;INITIALIZE THE TEST
 (1)  45DA  3E  0B              7.0                        MVI     A,@13      ;DEFINE THE TEST NUMBER
 (1)  45DC  CD  03  28         18.0                        CALL    TSET       ;SETUP THE TEST
2473                                  ;%RPM2-VELOCITY TEST-FROM TU PORT #2
2474                                  ;&M8953, M8955'S
2475
2476  45DF  3A  47  48         13.0  TST13X: LDA     UNITMP           ;GET THE UNIT MAP BYTE
2477  45E2  E6  04                    7.0         ANI     $04              ;TEST TU PORT #2
2478  45E4  CA  03  46         10.0         JZ      TEST14           ;NO-GO TO NEXT TEST
2479  45E7  CD  1E  47         18.0         CALL    CLEAR            ;CLEAR ALL TU PORTS
2480  45EA  3E  02                    7.0         MVI     A,@2
2481  45EC  CD  50  46         18.0         CALL    VELCTL
2482  45EF                               ENDTST  TST13X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45EF                          REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45EF  CD  06  28         18.0                   CALL    REQST
 (2)  45F2  00                                        .BYTE                    ;DATA PATTERN NUMBER
 (2)  45F3  00  00                                    .WORD                    ;SYSTEM '*' COUNT
 (2)  45F5  00  00                                    .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45F7  00                                        .BYTE                    ;DATA COMPARE FLAG IF -1
 (2)  45F8  07                                        .BYTE   7                ;REQUEST CODE
 (1)  45F9  3A  9A  4F         13.0         LDA     ITERA            ;GET ITERATION COUNT
 (1)  45FC  3D                   4.0         DCR     A                ;DOWNCOUNT
 (1)  45FD  32  9A  4F         13.0         STA     ITERA            ;SAVE COUNT
 (1)  4600  F2  DF  45         10.0         JP      TST13X           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - VELOCITY TEST - FROM TU PORT #3
                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*----- -----
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                                ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                                ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 3.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #3 SELECTED BY USER
                                ;*       THEN-CONTINUE
                                ;*       ELSE-NEXT TEST
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     LOAD THE PORT #3 CONTROL ADDRESS
                                ;*     CALL SUBROUTINE VELCTL
                                ;*ENDTST
                                SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 14
                                ;*RPM2 MICRO ERROR 04
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT -25%
                                ;*
                                ;*RPM2 MICRO TEST 14
                                ;*RPM2 MICRO ERROR 05
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
                                ;*M8953, M8955'S
                                ;*VELOCITY OK NOT SET AT NOMINAL
                                ;*
                                ;*RPM2 MICRO TEST 14
                                ;*RPM2 MICRO ERROR 06
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT +25%
                                S
                                ;*********************************************************************
```

```
2521  4603                                    TEST14: TESTX   @14              ;INITIALIZE THE TEST
 (1)  4603  3E  0C              7.0                      MVI     A,@14           ;DEFINE THE TEST NUMBER
 (1)  4605  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
2522                                    ;%RPM2-VELOCITY TEST-FROM TU PORT #3
2523                                    ;&M8953, M8955'S
2524  4608  3A  47  48          13.0    TST14X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2525  4608  E6  10                7.0            ANI     $10             ;TEST TU PORT #3
2526  460D  CA  2C  46           10.0            JZ      EXIT            ;NO-GO TO NEXT TEST
2527  4610  CD  1E  47           18.0            CALL    CLEAR           ;CLEAR ALL TU PORTS
2528  4613  3F  03                7.0            MVI     A,@3
2529  4615  CD  50  46           18.0            CALL    VELCTL
2530  4618                                      ENDTST  TST14X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4618                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4618  CD  06  28         18.0                      CALL    REQST
 (2)  461B  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  461C  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  461E  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4620  00                                           .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4621  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4622  3A  9A  4F          13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  4625  3D                  4.0             DCR     A               ;DOWNCOUNT
 (1)  4626  32  9A  4F          13.0            STA     ITERA           ;SAVE COUNT
 (1)  4629  F2  08  46          10.0            JP      TST14X          ;DO TEST UNTIL TILL = 0
2531
2532  462C  CD  1E  47          18.0    EXIT:   CALL    CLEAR           ;CLEAR ALL PORTS BEFORE LEAVING
2533  462F  C3  18  28          10.0            JMP     TSTEND          ;END OF TEST
```

```
                                .SBTTL  SUBROUTINE TACH PULSE GENERATOR
                               SSUB
                               ; ***************************************************************************
                               ;*SUBROUTINE TITLE
                               ;*---------- -----
                               ;*TACH PULSE GENERATOR
                               SD
                               ; ***************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS SUBROUTINE ISSUES 4 TACH PULSES FROM THE TAPE UNIT PORT AT A
                               ;*PULSE RATE CALCUALTED BY THE INTEGER IN MEMORY LOCATION ''TICNM'' TIMES
                               ;*5 MICROSECONDS.
                               SP
                               ; ***************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*    LOAD THE NUMBER OF TACH PULSES TO BE GENERATED
                               ;*    BGNDO
                               ;*       :    OUTPUT THE TACH BIT=1 FROM THE SPECIFIED TU PORT
                               ;*       :    OUTPUT THE TACH BIT=0 FROM THE SPECIFIED TU PORT
                               ;*       :    DELAY FOR THE TIME SPECIFIED BY ''TICNM'' X 5 MICRO SECONDS
                               ;*       :    DECREMENT THE TACH PULSE COUNTER
                               ;*       :    DO UNTIL THE TACH PULSE COUNTER = ZERO
                               ;*    ENDDO
                               ;*ENDTST
                               S
                               ; ***************************************************************************

                               ; ***************************************************************************

 4632 06 04          7.0  VELOK:  MVI   B,4              ;LOAD THE TACH TICK COUNT
 4634 3E 04          7.0          MVI   A,P.RPEN
 4636 D3 4C         10.0          OUT   PENAB
 4638 3E 28          7.0  1$:     MVI   A,P.LWR!P.TACH   ;LOAD THE TACH BIT
 463A D3 48         10.0          OUT   PDIAG            ;SET THE TACH BIT
 463C 3E 20          7.0          MVI   A,P.LWR          ;CLEAR THE TACH PULSE BIT
 463E D3 48         10.0          OUT   PDIAG

                               ;PERFORM A TIMEOUT

 4640 3A 4A 48      13.0          LDA   TICNM            ;GET THE TICK COUNTER

 4643 3D             4.0  2$:     DCR   A                ;WAIT UNTIL ZERO
 4644 C2 43 46      10.0          JNZ   2$               ;
                                                         ;DECREMENT THE TICK COUNT
 4647 05             4.0          DCR   B                ;CONTINUE UNTIL ALL
 4648 C2 38 46      10.0          JNZ   1$               ;TACKS ISSUED.
                                                         ;
 464B DB 01         10.0          IN    RPCHI            ;GET THE VELOCITY CHECK WORD
 464D E6 80          7.0          ANI   R.VOK            ;VELOCITY OK?
 464F C9            10.0          RET                    ;RETURN TO CALLER
```

```
                                .SBTTL  SUBROUTINE VELCTL
                               SSUB
                               ; ***************************************************************************
                               ;*SUBROUTINE TITLE
                               ;*---------- -----
                               ;*SUBROUTINE VELOCITY CONTROL
                               SD
                               ; ***************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS SUBROUTINE SIMULATES TACH PULSES FROM THE TAPE TRANSPORT AT -25%
                               ;*OF NOMINAL, NOMINAL (40 MICRO SECONDS), AND +25% OF NOMINAL.  THE VEL-
                               ;*OCITY OK STATUS BIT IS CHECKED AND THE FOLLOWING RESULTS ARE EXPECTED.
                               ;*FAILS (VELOCITY OK = 0) AT -25% OF NOMINAL
                               ;*PASSES (VELOCITY OK = 1) AT NOMINAL
                               ;*FAILS (VELOCITY OK = 0) AT +25% OF NOMINAL
                               SP
                               ; ***************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNSUB
                               ;*    LOAD THE 30 MICRO SECOND TICK RATE VARIABLE
                               ;*    CALL SUBROUTINE 'VELOK'
                               ;*      IF VELOCITY=OK
                               ;*      :   THEN-ERROR
                               ;*      :   ELSE-CONTINUE
                               ;*      ENDIF
                               ;*    LOAD THE 40 MICRO SECOND TICK RATE VARIABLE
                               ;*    CALL SUBROUTINE 'VELOK'
                               ;*      IF VELOCITY-OK
                               ;*      :   THEN-CONTINUE
                               ;*      :   ELSE-ERROR
                               ;*      ENDIF
                               ;*    LOAD THE 50 MICRO SECOND TICK RATE VARIABLE
                               ;*    CALL SUBROUTINE 'VELOK'
                               ;*      IF VELOCITY-OK
                               ;*      :   THEN-ERROR
                               ;*      :   ELSE-CONTINUE
                               ;*      ENDIF
                               ;*ENDSUB
                               S
                               ; ***************************************************************************

4650  47              4.0  VELCTL: MOV     B,A
4651  DB  E0         10.0         IN      INTSTA
4653  E6  80          7.0         ANI     BIT7
4655  B0              4.0         ORA     B
4656  D3  E0         10.0         OUT     MBSEL
4658  3E  02          7.0  TSTVLO: MVI     A,2             ;LOAD THE 30 MICRO SECOND TICK RATE
465A  32  4A  48      13.0         STA     TICNM           ;SAVE IT
465D  CD  32  46      18.0         CALL    VELOK           ;CALL THE VELOCITY ROUTINE
4660  CA  68  46      10.0         JZ      VELCNO          ;CONTINUE IF VELOCITY OK RESET
```

```
2620  4663                                          ERR     TSTVL0,VELCN0
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4663  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4666  04                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4667  00                                             .BYTE
 (1)  4668  CD  15  28        18.0                  VELCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  466B  DA  58  46        10.0                          JC      TSTVL0          ;LOOP ADDRESS IF LOOP SPECIFIED
2621                                          ;>VELOCITY OK SET AT -25%
2622  466E  3E  04               7.0  TSTVL1: MVI     A,4             ;LOAD A 40 MICRO SECOND TICK RATE
2623  4670  32  4A  48           13.0          STA     TICNM           ;SAVE IT
2624  4673  CD  32  46           18.0          CALL    VELOK           ;CALL THE VELOCITY ROUTINE
2625  4676  C2  7E  46           10.0          JNZ     VELCN1          ;CONTINUE IF VELOCITY OK SET
2626  4679                                    ERR     TSTVL1,VELCN1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4679  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  467C  05                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  467D  00                                             .BYTE
 (1)  467E  CD  15  28        18.0                  VELCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4681  DA  6E  46        10.0                          JC      TSTVL1          ;LOOP ADDRESS IF LOOP SPECIFIED
2627                                          ;>VELOCITY OK NOT SET AT NOMINAL
2628
2629
2630  4684  3E  07               7.0  TSTVL2: MVI     A,7             ;LOAD A 50 MICROSECOND TICK RATE
2631  4686  32  4A  48           13.0          STA     TICNM           ;SAVE IT
2632  4689  CD  32  46           18.0          CALL    VELOK           ;CALL VELOCITY ROUTINE
2633  468C  CA  94  46           10.0          JZ      VELCN2          ;CONTINUE IF VELOCITY OK RESET
2634  468F                                    ERR     TSTVL2,VELCN2
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  468F  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                                            MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4692  06                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4693  00                                             .BYTE
 (1)  4694  CD  15  28        18.0                  VELCN2::        CALL    CK              ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4697  DA  84  46        10.0                          JC      TSTVL2          ;LOOP ADDRESS IF LOOP SPECIFIED
2635                                          ;>VELOCITY OK SET AT +25%
2636  469A  C9                  10.0          RET
2637
2638
```

```
                                .SBTTL  SUBROUTINE SET UP THE READ PATH FOR FIFO TEST
4698                            SSUB
 (1)                            ;*********************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*SET UP READ PATH FOR FIFO TEST - FROM PORT
4698                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE SETS THE READ CHANNELS UP FOR THE FIFO DATA TEST
                                ;*FROM THE PORTS.
4698                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     SET PLO BYPASS MODE 2
                                ;*     CLOCK THE FIFO'S
                                ;*     ISSUE CLEAR ALL FIFO'S COMMAND
                                ;*     WAIT 5 MICROSECONDS
                                ;*     STOP THE READ PATH AND READ CHANNELS
                                ;*     CLEAR THE EXPECTED DATA
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     DISABLE THE TRANSLATOR OUTPUTS
                                ;*     ISSUE TU PORT CONTROL COMMAND (DPEN, LCS, LWR)
                                ;*     BGNDO
                                ;*        :     WRITE EXPECTED DATA TO THE READ DATA LINES
                                ;*        :     CALCULATE THE PARITY BIT FOR THE EXPECTED DATA
                                ;*        :     WRITE EXPECTED DATA PARITY TO THE READ DATA LIENS
                                ;*        :     CALL SUBROUTINE RDPLO2
                                ;*        :     CALL SUBROUTINE CKAMT
                                ;*        :     INCREMENT THE EXPECTED DATA BYTE
                                ;*        :     DO UNTIL THE EXPECTED DATA BYTE-ZERO
                                ;*     ENDDO
                                ;*ENDSUB
4698                            S
 (1)                            ;*********************************************************************

469B  47            4.0  RDSET:  MOV     B,A
469C  3E  10        7.0          MVI     A,RDCLK         ;GET NORMAL READ PATH CLOCK
469E  D3  F0       10.0          OUT     CLKCTL          ;SET UP THE CLOCK
46A0  3E  20        7.0          MVI     A,R.PLO1        ;GET PLO BYPASS MODE 2
46A2  D3  09       10.0          OUT     RPCTL           ;SET PLO BYPASS

46A4  D3  08       10.0          OUT     RFIFOL          ;CLOCK THE FIFO
46A6  3E  0D        7.0          MVI     A,RCLRT         ;LOAD ALL FIFO'S CLEAR COMMAND
46A8  D3  08       10.0          OUT     RCMD            ;ISSUE FIFO CLEAR

46AA  00            4.0          NOP                     ;WAIT
46AB  00            4.0          NOP                     ;WAIT
46AC  00            4.0          NOP                     ;WAIT
46AD  00            4.0          NOP                     ;WAIT
```

```
2684
2685  46AE  3E  21        7.0          MVI   A,R.PLO1!R.STPC ;STOP THE RMC SO THAT
2686  46B0  D3  09       10.0          OUT   RPCTL           ;THE FIFO CAN BE LOADED
2687  46B2  DB  E0       10.0          IN    INTSTA          ;
2688  46B4  E6  80        7.0          ANI   BIT7            ;
2689  46B6  B0            4.0          ORA   B               ;
2690  46B7  D3  E0       10.0          OUT   MBSEL           ;
2691  46B9  AF            4.0          XRA   A               ;CLEAR THE ACCUMULATOR
2692  46BA  32  43  48   13.0          STA   DATAE           ;STORE AS EXPECTED DATA
2693  46BD  D3  D2       10.0          OUT   TR.ENA          ;DISABLE THE TRANSLATOR OUTPUTS
2694  46BF  D3  D2       10.0          OUT   TRKENA
2695  46C1                             ROUT  R05H
 (1)  46C1  D3  8B       10.0                OUT   R05H            ;WRITE AC INTO R05H
 (1)  46C3  7F            4.0                MOV   A,A             ;RETRY LINK
2696  46C4  3E  60        7.0          MVI   A,P.LCS.P.LWR
2697  46C6  D3  48       10.0          OUT   PDIAG           ;LOOP THE 8085 TO READ PATH
2698  46C8  3E  04        7.0          MVI   A,P.RPEN
2699  46CA  D3  4C       10.0          OUT   PENAB
2700
2701  46CC  3A  43  48   13.0  3$:     LDA   DATAE           ;GET THE EXPECTED DATA
2702  46CF  D3  40       10.0          OUT   TCMD            ;PUT ON THE READ DATA LINES
2703  46D1                             ROUT  R05L            ;WRITE THE EXPECTED DATA TO CAS REGISTER 5
 (1)  46D1  D3  8A       10.0                OUT   R05L            ;WRITE AC INTO R05L
 (1)  46D3  7F            4.0                MOV   A,A             ;RETRY LINK
2704
2705                                 ;CALCULATE EXPECTED DATA PARITY
2706  46D4  A7            4.0          ANA   A               ;SET CONDITION BITS
2707  46D5  E2  DD  46   10.0          JPO   1$              ;ODD NUMBER OF ONES
2708  46D8  3E  01        7.0          MVI   A,$01           ;ELSE-EVEN NUMBER OF ONES
2709  46DA  C3  DE  46   10.0          JMP   2$              ;PARITY BIT SHOULD BE ONE
2710
2711  46DD  AF            4.0  1$:     XRA   A               ;PARITY BIT SHOULD BE ZERO
2712  46DE  32  44  48   13.0  2$:     STA   DATAEP          ;SAVE EXPECTED DATA PARITY
2713  46E1  07            4.0          RLC                   ;SHIFT THE PARITY BIT TO THE RIGHT PLACE
2714  46E2  F6  60        7.0          ORI   P.LCS!P.LWR
2715  46E4  D3  48       10.0          OUT   PDIAG
2716
2717  46E6  CD  FA  46   18.0          CALL  RDPL02          ;GO CLOCK DATA THROUGH THE FIFO
2718  46E9  CD  49  47   18.0          CALL  CKAMT           ;GO CHECK FOR BAD READ CHANNELS
2719  46EC  DA  CC  46   10.0          JC    3$              ;LOOP ON ERROR
2720
2721  46EF  3A  43  48   13.0          LDA   DATAE           ;GET EXPECTED DATA
2722  46F2  3C            4.0          INR   A               ;UPDATE THE EXPECTED DATA
2723  46F3  32  43  48   13.0          STA   DATAE           ;SAVE EXPECTED DATA
2724  46F6  C2  CC  46   10.0          JNZ   3$              ;CONTINUE UNTIL DONE
2725
2726  46F9  C9           10.0          RET
2727
2728
```

```
                                        .SBTTL  SUBROUTINE READ THE FIFO'S
46FA                                    SSUB
 (1)                                    ; ***************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*READ THE FIFO'S SUBROUTINE
46FA                                    SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*BGNSUB
                                        ;*    CLOCK THE DATA FROM THE TU PORT INTO THE FIFO'S
                                        ;*    PUT THE TM78 SUBSYSTEM IN READ PE MODE
                                        ;*    ISSUE THE FIFO READ COMMAND
                                        ;*    ENABLE THE READ PATH AND READ CHANNEL CLOCK
                                        ;*    DELAY 5 MICROSECONDS
                                        ;*    DISABLE THE READ PATH AND READ CHANNEL CLOCK
                                        ;*    READ THE FIFO DATA (9 BITS)
                                        ;*ENDSUB
46FA                                    S
 (1)                                    ; ***************************************************************************

46FA  D3  08          10.0  RDPL02: OUT     RFIFOL          ;CLOCK THE DATA INTO THE FIFO

46FC  AF               4.0          XRA     A               ;CLEAR A
46FD  D3  D3          10.0          OUT     WMCCTL          ;PUT THE MACHINE IN PE MODE

46FF  3E  6A           7.0          MVI     A,FIFORD        ;LOAD THE FIFO READ COMMAND
4701  D3  0B          10.0          OUT     RCMD

4703  3E  20           7.0          MVI     A,R.PL01        ;LET THE READ MACHINE RUN
4705  D3  09          10.0          OUT     RPCTL

4707  00               4.0          NOP                     ;WAIT
4708  00               4.0          NOP                     ;WAIT
4709  00               4.0          NOP                     ;WAIT
470A  00               4.0          NOP                     ;WAIT

470B  3E  21           7.0          MVI     A,R.PL01!R.STPC ;STOP THE READ MACHINE
470D  D3  09          10.0          OUT     RPCTL

470F  DB  17          10.0          IN      RDATA           ;READ THE FIFO DATA
4711  32  45  48      13.0          STA     DATAA           ;STORE ACTUAL DATA
4714  DB  15          10.0          IN      RPSTA           ;READ THE FIFO DATA PARITY
4716  E6  40           7.0          ANI     R.DATA          ;REMOVE UNWANTED BITS
4718  07               4.0          RLC                     ;JUSTIFY THE ACTUAL
4719  07               4.0          RLC                     ;DATA PARITY
471A  32  46  48      13.0          STA     DATAAP          ;STORE ACTUAL DATA PARITY
471D  C9              10.0          RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
471E                                    SSUB
                                        ; *******************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
471E                                    SD
                                        ; *******************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
471E                                    SP
                                        ; *******************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
471E                                    S
                                        ; *********************************************************************
471E  F5                  12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
471F  C5                  12.0          PUSH    B
4720  06    00             7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
4722  DB    E0            10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4724  E6    80             7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4726  B0                   4.0          ORA     B               ;ADD IN THE SELECTED PORT #
4727  D3    E0            10.0          OUT     MBSEL           ;RESET TO THIS PORT
4729  3E    80             7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
472B  D3    40            10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
472D  AF                   4.0          XRA     A               ;CLEAR TU COMMAND A
472E  D3    40            10.0          OUT     TCMD            ;
4730  3E    81             7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4732  D3    40            10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4734  3E    00             7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
4736  D3    40            10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
4738  AF                   4.0          XRA     A
4739  D3    44            10.0          OUT     TAMT            ;CLEAR AMTIE WORD
473B  D3    48            10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
473D  D3    4C            10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
473F  04                   4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
4740  78                   4.0          MOV     A,B
4741  FE    04             7.0          CPI     4               ;DONE?
4743  C2    22    47      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
4746  C1                  10.0          POP     B               ;RESET B & C
4747  F1                  10.0          POP     PSW             ;ALL DONE
4748  C9                  10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE CHECK AMTIE
                               SSUB
                               ;*****************************************************************************
                               ;*SUBROUTINE TITLE
                               ;*---------- -----
                               ;*CHECK AMTIE SUBROUTINE
                               SD
                               ;*****************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ PATH DATA
                               ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
                               ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
                               SP
                               ;*****************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNSUB
                               ;*     IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
                               ;*       THEN-ERROR CHANNEL 0
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
                               ;*       THEN-ERROR CHANNEL 1
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
                               ;*       THEN-ERROR CHANNEL 2
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL 3 ACTUAL INFORMATION NOT - CHANNEL 3 EXPECTED
                               ;*       THEN-ERROR CHANNEL 3
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
                               ;*       THEN-ERROR CHANNEL 4
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
                               ;*       THEN-ERROR CHANNEL 5
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
                               ;*       THEN-ERROR CHANNEL 6
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL 7 ACTUAL INFORMATION NOT - CHANNEL 7 EXPECTED
                               ;*       THEN-ERROR CHANNEL 7
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
                               ;*     IF CHANNEL P ACTUAL INFORMATION NOT - CHANNEL P EXPECTED
                               ;*       THEN-ERROR CHANNEL P
                               ;*       ELSE-CONTINUE
                               ;*     ENDIF
```

```
2864
2865  4749                                          ;*ENDSUB
 (1)                                                S
                                                    ; ***************************************************************************
2866  4749  21  43  48        10.0  CKAMT:  LXI     H,DATAE             ;LOAD EXPECTED DATA ADDRESS
2867  474C  3A  45  48        13.0          LDA     DATAA               ;GET THE ACTUAL DATA
2868  474F                         .       ROUT    ADATA               ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  474F  D3  94              10.0                  OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  4751  7F                   4.0                  MOV     A,A                 ;RETRY LINK
2869  4752  E6  01               7.0          ANI     $01                 ;REMOVE UNWANTED BITS
2870  4754  47                   4.0          MOV     B,A                 ;SAVE IN B
2871  4755  7E                   7.0          MOV     A,M                 ;GET EXPECTED DATA
2872  4756                                     ROUT    EDATA               ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  4756  D3  95              10.0                  OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  4758  7F                   4.0                  MOV     A,A                 ;RETRY LINK
2873  4759  E6  01               7.0          ANI     $01                 ;REMOVE UNWANTED BITS
2874  475B  B8                   4.0          CMP     B                   ;COMPARE
2875  475C  CA  64  47         10.0          JZ      CKAM0               ;CONTINUE IF EQUAL
2876  475F                                     ERRB    OUT,CKAM0
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  475F  CD  12  28         18.0                  CALL    ERLPB               ;PROCESS ERROR - DO 2.3
 (1)        0007                                    MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4762    07                                    .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  4763    00                                    .BYTE                       ;PRINT ROUTINE NUMBER
 (1)  4764  CD  15  28         18.0          CKAM0:: CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4767  DA  2D  48         10.0                  JC      OUT                 ;LOOP ADDRESS IF LOOP SPECIFIED
2877                                     ;>CHANNEL 0 FAILED
2878
2879  476A  3A  45  48        13.0          LDA     DATAA               ;GET THE ACTUAL DATA
2880  476D  E6  02               7.0          ANI     $02                 ;REMOVE UNWANTED BITS
2881  476F  47                   4.0          MOV     B,A                 ;SAVE IN B
2882  4770  7E                   7.0          MOV     A,M                 ;GET EXPECTED DATA
2883  4771  E6  02               7.0          ANI     $02                 ;REMOVE UNWANTED BITS
2884  4773  B8                   4.0          CMP     B                   ;COMPARE
2885  4774  CA  7C  47         10.0          JZ      CKAM1               ;CONTINUE IF EQUAL
2886  4777                                     ERRB    OUT,CKAM1
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4777  CD  12  28         18.0                  CALL    ERLPB               ;PROCESS ERROR - DO 2.3
 (1)        0008                                    MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  477A    08                                    .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  477B    00                                    .BYTE                       ;PRINT ROUTINE NUMBER
 (1)  477C  CD  15  28         18.0          CKAM1:: CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  477F  DA  2D  48         10.0                  JC      OUT                 ;LOOP ADDRESS IF LOOP SPECIFIED
2887                                     ;>CHANNEL 1 FAILED
```

```
2889
2890
2891  4782  3A 45 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2892  4785  E6 04           7.0          ANI     $04             ;REMOVE UNWANTED BITS
2893  4787  47              4.0          MOV     B,A             ;SAVE IN B
2894  4788  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2895  4789  E6 04           7.0          ANI     $04             ;REMOVE UNWANTED BITS
2896  478B  B8              4.0          CMP     B               ;COMPARE
2897  478C  CA 94 47       10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
2898  478F                               ERRB    OUT,CKAM2
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  478F  CD 12 28   18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0009                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4792  09                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4793  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4794  CD 15 28   18.0                CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4797  DA 2D 48   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2899                                ;>CHANNEL 2 FAILED
2900
2901  479A  3A 45 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2902  479D  E6 08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
2903  479F  47              4.0          MOV     B,A             ;SAVE IN B
2904  47A0  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2905  47A1  E6 08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
2906  47A3  B8              4.0          CMP     B               ;COMPARE
2907  47A4  CA AC 47       10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2908  47A7                               ERRB    OUT,CKAM3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  47A7  CD 12 28   18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000A                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47AA  0A                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47AB  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47AC  CD 15 28   18.0                CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47AF  DA 2D 48   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2909                                ;>CHANNEL 3 FAILED
2910
2911
2912  47B2  3A 45 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2913  47B5  E6 10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
2914  47B7  47              4.0          MOV     B,A             ;SAVE IN B
2915  47B8  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2916  47B9  E6 10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
2917  47BB  B8              4.0          CMP     B               ;COMPARE
2918  47BC  CA C4 47       10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2919  47BF                               ERRB    OUT,CKAM4
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  47BF  CD 12 28   18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47C2  0B                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47C3  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47C4  CD 15 28   18.0                CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47C7  DA 2D 48   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2920                                ;>CHANNEL 4 FAILED
```

```
2922
2923  47CA  3A 45 48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2924  47CD  E6 20       7.0          ANI     $20             ;REMOVE UNWANTED BITS
2925  47CF  47          4.0          MOV     B,A             ;SAVE IN B
2926  47D0  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2927  47D1  E6 20       7.0          ANI     $20             ;REMOVE UNWANTED BITS
2928  47D3  B8          4.0          CMP     B               ;COMPARE
2929  47D4  CA DC 47    10.0         JZ      CKAM5           ;CONTINUE IF EQUAL
2930  47D7                           ERRB    OUT,CKAM5
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  47D7  CD 12 28   18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000C                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47DA  0C                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47DB  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47DC  CD 15 28   18.0           CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47DF  DA 2D 48   10.0                     JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2931                            ;>CHANNEL 5 FAILED
2932
2933
2934  47E2  3A 45 48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2935  47E5  E6 40       7.0          ANI     $40             ;REMOVE UNWANTED BITS
2936  47E7  47          4.0          MOV     B,A             ;SAVE IN B
2937  47E8  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2938  47E9  E6 40       7.0          ANI     $40             ;REMOVE UNWANTED BITS
2939  47EB  B8          4.0          CMP     B               ;COMPARE
2940  47EC  CA F4 47    10.0         JZ      CKAM6           ;CONTINUE IF EQUAL
2941  47EF                           ERRB    OUT,CKAM6
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  47EF  CD 12 28   18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000D                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47F2  0D                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47F3  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47F4  CD 15 28   18.0           CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47F7  DA 2D 48   10.0                     JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2942                            ;>CHANNEL 6 FAILED
2943
2944  47FA  3A 45 48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2945  47FD  E6 80       7.0          ANI     $80             ;REMOVE UNWANTED BITS
2946  47FF  47          4.0          MOV     B,A             ;SAVE IN B
2947  4800  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2948  4801  E6 80       7.0          ANI     $80             ;REMOVE UNWANTED BITS
2949  4803  B8          4.0          CMP     B               ;COMPARE
2950  4804  CA 0C 48    10.0         JZ      CKAM7           ;CONTINUE IF EQUAL
2951  4807                           ERRB    OUT,CKAM7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4807  CD 12 28   18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000E                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  480A  0E                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  480B  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  480C  CD 15 28   18.0           CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  480F  DA 2D 48   10.0                     JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2952                            ;>CHANNEL 7 FAILED
```

```
2954
2955
2956  4812  21  44  48      10.0          LXI     H,DATAEP        ;GET THE EXPECTED DATA PARITY ADDRESS
2957  4815  3A  46  48      13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
2958  4818  E6  01           7.0          ANI     $01             ;REMOVE UNWANTED BITS
2959  481A  47               4.0          MOV     B,A             ;SAVE IN B
2960  481B  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
2961  481C  E6  01           7.0          ANI     $01             ;REMOVE UNWANTED BITS
2962  481E  B8               4.0          CMP     B               ;COMPARE
2963  481F  CA  27  48       10.0         JZ      CKAMP           ;CONTINUE IF EQUAL
2964  4822                                ERRB    OUT,CKAMP,1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4822  CD  12  28       18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000F                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4825  0F                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4826  01                                     .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4827  CD  15  28       18.0          CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  482A  DA  2D  48       10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2965                                    ;>CHANNEL P FAILED
2966
2967  482D  C9                10.0    OUT:    RET
```

```
2969                                    .SBTTL  TABLE EXPECTED FIFO DATA
2970
2971  482E  00          FIFOEX: .BYTE   $00          ;FIFO READ 01 DATA
2972  482F  00                  .BYTE   $00          ;FIFO READ 01 PARITY
2973  4830  FF                  .BYTE   $FF          ;FIFO READ 02 DATA
2974  4831  01                  .BYTE   $01          ;FIFO READ 02 PARITY
2975  4832  00                  .BYTE   $00          ;FIFO READ 03 DATA
2976  4833  00                  .BYTE   $00          ;FIFO READ 03 PARITY
2977  4834  FF                  .BYTE   $FF          ;FIFO READ 04 DATA
2978  4835  01                  .BYTE   $01          ;FIFO READ 04 PARITY
2979  4836  00                  .BYTE   $00          ;FIFO READ 05 DATA
2980  4837  00                  .BYTE   $00          ;FIFO READ 05 PARITY
2981  4838  FF                  .BYTE   $FF          ;FIFO READ 06 DATA
2982  4839  01                  .BYTE   $01          ;FIFO READ 06 PARITY
2983  483A  00                  .BYTE   $00          ;FIFO READ 07 DATA
2984  483B  00                  .BYTE   $00          ;FIFO READ 07 PARITY
2985  483C  FF                  .BYTE   $FF          ;FIFO READ 08 DATA
2986  483D  01                  .BYTE   $01          ;FIFO READ 08 PARITY
2987  483E  00                  .BYTE   $00          ;FIFO READ 09 DATA
2988  483F  00                  .BYTE   $00          ;FIFO READ 09 PARITY
2989  4840  FF                  .BYTE   $FF          ;FIFO READ 10 DATA
2990  4841  01                  .BYTE   $01          ;FIFO READ 10 PARITY
2991  4842  00          FIFOEN: .BYTE   0            ;END OF TABLE
```

```
                                        .SBTTL  PROGRAM VARIABLES

                                DATAE:  .BYTE   0               ;EXPECTED DATA BYTE
                                DATAEP: .BYTE   0               ;EXPECTED DATA BYTE PARITY
                                DATAA:  .BYTE   0               ;ACTUAL DATA BYTE
                                DATAAP: .BYTE   0               ;ACTUAL DATA BYTE PARITY
                                UNITMP: .BYTE   0               ;UNITS MAP
                                PASCT:  .BYTE   0               ;PASS COUNTER
                                PASFL:  .BYTE   0               ;FAILING PASS NUMBER
                                TICNM:  .BYTE   0               ;TACK RATE COUNTER

                                        .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKAMP   4827  G
CKAMT   4749           CKAM0   4764  G        CKAM1   477C  G       CKAM2   4794  G
CKAM3   47AC  G        CKAM4   47C4  G        CKAM5   47DC  G       CKAM6   47F4  G
CKAM7   480C  G        CKLOP  = 2815          CLEAR   471E          CLKCTL= 00F0
CLOCK   4F26           CLRLP   4722           CMC0H  = 0099          CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CRCWRD= 0018           CSAVE   4F9D          CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE
C.MAIN- 0020           C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002
D      =%0002          DATAA   4845           DATAAP  4846          DATACT= 00D0
DATAE   4843           DATAEP  4844           DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL- 00DB           DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 0008          DONE1  = 0045          DONINT= 0010          DSAVE   4F9E
DSE    = 0006          DUMMY   4490  G        D.ATH0= 0001          D.ATH1= 0002
D.EOTD- 0010           D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041          ECCSTA- 001A          ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003           ERFLG   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812          ERLPE  = 280C         ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F           EXIT    462C          E.ACRC= 0010
E.AMT  - 0020          E.CDP  = 0080          E.CRC  = 0080         E.PNTR= 0008
E.RPE  - 0040          E.STEC= 0001           E.TTEC= 0002          E.UNC  = 0004
FIFOEN  4842           FIFOEX  482E           FIFOL1  4309          FIFOL2  436B
FIFORD= 006A           FORMAT  4F25           FWDTST= 0061          GCRID  = 0089
GCRSET= 0002           GOODTM= 0092           H      =%0004          HLSAVE  4FA0
IE     = 0008          INTSTA= 00E0           ITERA   4F9A          I.PWR  = 0020
I.RMPE  0040           I5.5   = 0010          I6.5   = 0020         I7.5   = 0040
KCALL  = 005F          KCLR   = 007B          KDEP   = 003F         KENAB  = 0078
KEXAM  = 003E          KEYBRD- 00C8           KEY1   - 0078         KEY10  = 006D
KEY11  = 006E          KEY12  = 006F          KEY13  = 005C         KEY14  = 005D
KEY15  = 005E          KEY16  = 005F          KEY17  = 003C         KEY18  = 003D
KEY19  - 003E          KEY2   = 0079          KEY20  = 003F         KEY3   = 007A
KEY4   - 007B          KEY5   - 0074          KEY6   = 0075         KEY7   - 0076
```

```
KEY8   = 0077    KEY9   = 006C    KINTA  = 006F    KLDAD  = 003D
KN0    = 003C    KN1    = 005C    KN2    = 005D    KN3    = 005E
KN4    = 006C    KN5    = 006D    KN6    = 006E    KN7    = 0074
KN8    = 0075    KN9    = 0076    KU2    = 0079    KU3    = 007A
KU8    = 0077    L      =%0005    LBLANK= 000F     LCE    = 000B
LCH    = 000C    LCL    = 000D    LCP    = 000E    LC0    = 0000
LC1    = 0001    LC2    = 0002    LC3    = 0003    LC4    = 0004
LC5    = 0005    LC6    = 0006    LC7    = 0007    LC8    = 0008
LC9    = 0009    LDLEDA= 00CA     LDLEDB= 00CB     LDLEDC= 00CC
LDLEDD= 00CD     LDLEDE= 00CE     LDLEDF= 00CF     LKDIAG= 2800
LKKBD  = 004C    LKKEY  = 0049    LKLWMG= 0058     LKLWMP= 0055
LKLWPG= 0052     LKLWPP= 004F     LKMOD7= 0046     LKOPR  = 0046
LPFLG   4F94     LPNUM   4F92     M      =%0006    MBSEL  = 00E0
MB.A   = 0008    MB.B   = 0004    MEMTOP= 4FFF     MINUS  = 000A
MM     = 8000    MSE    = 0008    MSGN   = 000F     MTACLR= 0000
MT.ARA= 0020     MT.CPE= 0080     MT.DSE= 0001     MT.FWD= 0040
MT.INH= 0008     MT.LWR= 0004     MT.MOT= 0002     MT.NWT= 0080
MT.PEC= 0040     MT.PSB= 0004     MT.PS0= 0001     MT.PS1= 0002
MT.REV= 0020     MT.WRT= 0010     MT.Z   = 0008     M.ATA  = 0080
M.CAPE= 0020     M.CONT= 0080     M.DEM  = 0020     M.EBL  = 0010
M.EXC  = 0008    M.FAIL= 0008     M.ILR  = 0010     M.INIT= 0010
M.OCC  = 0020    M.ONLI= 0001     M.PE   = 0040     M.PORT= 0080
M.RDEN= 0002     M.RDPE= 0008     M.RUN  = 0004     M.SCLK= 0001
M.TRA  = 0040    M.UNIT= 0007     M.WCLK= 0040     M.WCLN= 0080
M.WREN= 0080     M5.5   = 0001    M6.5   = 0002     M7.5   = 0004
NOTCAP= 0088     OKAY   = 00FF    OPRRAM= 4300     OPSTRT= 0058
OPVER  = 0040    OUT     482D     PADCNT= 00D5     PADCRC= 0080
PASCT   4848     PASFL   4849     PDIAG  = 0048     PEID   = 008A
PENAB  = 004C    PESET  = 0001    PL     = 00B1     PRDD   = 004C
PRENF  = 009C    PS     = 00B2    PSTAT  = 0048     PSW    =%0009
P.AMTP= 0001     P.BCTC= 0040     P.CMDP= 0020     P.INTE= 0080
P.LCS  = 0040    P.LWR  = 0020    P.RDP  = 0002     P.RPEN= 0004
P.RPST= 0002     P.RP0E= 0020     P.RP1E= 0010     P.RP2E= 0020
P.RP3E= 0010     P.SING= 0080     P.STAT= 0002     P.STPE= 0080
P.TACH= 0008     P.TUPR= 0010     P.WCSP= 0004     P.WDS  = 0040
P.WFLP= 0001     P.WPEN= 0010     P.WP0E= 0008     P.WP1E= 0004
P.WP2E= 0008     P.WP3E= 0004     P.5VOK= 0002     QUE    = 281B
QUEM   = 281E    RAMT   = 0010    RARA   = 0006     RARAI  = 0004
RCHBD0= 0048     RCHBD1= 0047     RCHOK  = 0046     RCHTST= 000C
RCLRT  = 000D    RCMD   = 0008    RCMLP  = 0003     RCONT  = 0080
RDATA  = 0017    RDCLK  = 0010    RDON   = 0011     RDPLO?  46FA
RDSET   4698     READG  = 0007    REND   = 0014     REQST  = 2806
RESCHR= 00D1     REVTST= 0064     REWIND= 0004     RFIFOL= 0008
RGCLK  = 0002    RGCRI  = 0003    RIBG   = 0001     RILL   = 0012
RINST  = 000C    RMCTST= 0008     RMK2   = 0013     RNOP   = 0000
RPATH  = 0001    RPBAD  = 0044    RPCHI  = 0001     RPCLK  = 0003
RPCTI  = 0009    RPEI   = 0002    RPFAIL= 0000     RPF1   = 009D
RPF2   = 009E    RPOK   = 0043    RPOSTN= 0016     RPSTA  = 0015
RRCMT  = 000A    RSTAT  = 0002    RTIEB  = 000A     RTIER  = 0030
RTM    = 0005    RUNKI  = 0009    RUPTST= 005E     RWDUNL= 0005
R.AMT  = 0001    R.BOP  = 0008    R.DATA= 0040     R.DON  = 0002
R.DRDY= 0010     R.END  = 0010    R.ILL  = 0004     R.JVOK= 0004
R.MK2  = 0008    R.PLOD= 0008     R.PL00= 0010     R.PL01= 0020
R.POST= 0020     R.STNM= 0002     R.STOP= 0004     R.STPC= 0001
```

RPM2 - READ PATH MICROCONTROLLER PART #2 CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:03 PAGE 1-51 SEQ 0736
RPM2.M80 SYMBOL TABLE

```
R.TBJN= 0080        R.TSTD= 0040        R.VOK  = 0080        R00H   = 0081
R00L  = 0080        R01H  = 0083        R01L   = 0082        R02H   = 0085
R02L  = 0084        R03H  = 0087        R03L     0086        R04H   = 0089
R04L  = 0088        R05H  = 0088        R05L   = 008A        R06H   = 008D
R06L  = 008C        R07H  = 008F        R07L   = 008E        R10H   = 0091
R10L  = 0090        R11H  = 0093        R11L   = 0092        R12H   = 0095
R12L  = 0094        R13H  = 0097        R13L   = 0096        R14H   = 0099
R14L  = 0098        R15H  = 0098        R15L   = 009A        R16H   = 009D
R16L  = 009C        R17H  = 009F        R17L   = 009E        R7.5   = 0010
SELCLR= 0000        SETATA= 00A1        SID    = 0030        SOD    = 0080
SOE   = 0040        SP    =%0008        SSCLK  = 0040        SSTEP  = 0005
STACK = 4FFF        STATRM= 4F20        STPCT    4F20        STRSP  = 5000
TADR00= 0080        TADR01= 0081        TADR02= 0082        TADR03= 0083
TADR04= 0084        TADR05= 0085        TADR06= 0086        TADR07= 0087
TADR10= 0088        TADR11= 0089        TADR12= 008A        TADR13= 008B
TAMT  = 0044        TASEL = 0080        TCMD   = 0040       TC.FWD= 0040
TC.INH= 0008        TC.LWR= 0004        TC.REV= 0020        TC.WRT= 0010
TEMP    4F99        TEST1   4300        TEST10   4538       TEST11   4588
TEST12  45B1        TEST13  45DA        TEST14   4603       TEST2    4362
TEST3   43C2        TEST4   4472        TEST5    44BD       TEST6    44E6
TEST7   450F        TICNM   484A        TMF    = 0099       TMRDY  = 0040
TRKENA= 00D2        TSET  = 2803        TSTEND= 2818        TSTS   = 0040
TSTVL0  4658        TSTVL1  466E        TSTVL2   4684       TST01X   4305
TST02X  4367        TST03X  43C7        TST04X   4499       TST05X   44C2
TST06X  44EB        TST07X  4514        TST10X   453D       TST11X   458D
TST12X  45B6        TST13X  45DF        TST14X   4608       TUSEL0= 00D1
TUSEL1= 00D2        TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002
T.BOT = 0004        T.EOT = 0002        T.FPT  = 0001       T.NTHR= 0004
T.ONL = 0020        T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008
T.PS1J= 0010        T.RDY = 0080        T.RDYO= 0040        T.RWD  = 0010
T.SCLK= 0002        UIBG  = 00A1        UNITMP   4847       VALFC    4F98
VALTB   4F95        VELCN0   4668  G    VELCN1   467E  G     VELCN2   4694  G
VELCTL  4650        VELOK    4632       VELTST= 005B        VOKCN0   4552  G
VOKCN1  456E  G     VOKLP0   4541       VOKLP1   4558       WDR.P  = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT= 00D3        W.ACRC= 0004        W.CRC  = 0008       W.DIAG= 0002
W.DONN= 0040        W.ECC  = 0010       W.ENAB= 0080        W.ERR  = 0020
W.FMT  = 0070       W.GCR  = 0010       W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV  = 0004       W.ROME= 0010        W.RST  = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X      =%000A
X.DONN= 0080        X.ENAB   0040       X.PEPE= 0002        X.ROME= 0001
X.WCLK= 0001        Y      =%000B       .      = 484B
```

ERRORS DETECTED: 0

*RPM2.A78/PTP,RPM2-NLIST,PARAM,MACRO,LIST,RPM2
RUN-TIME: 5 7 0 SECONDS
CORE USED: 10K

```
                                .TITLE  RPM3 - READ PATH MICRO CONTROLLER PART #3
                                .SBTTL  TEST 01 - IBG DETECT TEST
                                ;ID     RPM3-READ PATH MICRO CONTROLLER PART #3

4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM - IBG DETECT TEST
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST IS DESIGNED TO CHECK THE READ PATHS ABILITY TO DETECT AN
                                ;*INTERBLOCK GAP AND AN INTERBLOCK GAP FAILURE.
4300                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  SET NORMAL READ PATH CLOCK
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  ENABLE THE READ MICRO CONTROLLER
                                ;*  INIT THE AMTIE MASK TO ZERO (9 BITS WIDE)
                                ;*  BGNDO
                                ;*      :   SET ALL 9 AMTIES FROM THE USER SPECIFIED PORT
                                ;*      :   DELAY 50 MICROSECONDS
                                ;*      :   ISSUE THE IBG COMMAND
                                ;*      :   DELAY 50 MICROSECONDS
                                ;*      :   READ THE READ MICRO STATUS
                                ;*      :       IF STATUS = 0
                                ;*      :       :   THEN-CONTINUE
                                ;*      :       :   ELSE-ERROR
                                ;*      :       ENDIF
                                ;*      :   ISSUE THE AMTIE LINE AS INDICATED BY THE AMTIE MASK
                                ;*      :   DELAY 50 MICROSECONDS
                                ;*      :   READ THE READ MICRO STATUS
                                ;*      :       IF STATUS = 'BADST'
                                ;*      :       :   THEN-CONTINUE
                                ;*      :       :   ELSE-ERROR
                                ;*      :       ENDIF
                                ;*      :   INCREMENT THE AMTIE MASK BY 1
                                ;*      :   DO UNTIL AMTIE MASK = 777(8)
                                ;*  ENDDO
                                ;*ENDTST
4300                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM3 MICRO TEST 01
                                ;*RPM3 MICRO ERROR 01
                                ;*RPM3-READ PATH-IBG DETECT
                                ;*M8953, M8950'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
```

```
1371                                    ;*FATAL ERROR - TEST ABORTED
1372                                    ;*
1373                                    ;*RPM3 MICRO TEST 01
1374                                    ;*RPM3 MICRO ERROR 02
1375                                    ;*RPM3-READ PATH-IBG DETECT
1376                                    ;*M8953, M8950'S
1377                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD IBG
1378                                    ;*ACTUAL = NNNN
1379                                    ;*EXPECTED = NNNN
1380                                    ;*
1381                                    ;*RPM3 MICRO TEST 01
1382                                    ;*RPM3 MICRO ERROR 03
1383                                    ;*RPM3-READ PATH-IBG DETECT
1384                                    ;*M8953, M8950'S
1385                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD IBG
1386                                    ;*ACTUAL = NNNN
1387                                    ;*EXPECTED = NNNN
1388                                    ;*EXPECTED 18 BITS =    E EEEEEE
1389                                    ;*ACTUAL 18 BITS =      A AAAAAA
1390                                    ;*EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1391  4300                              S
 (1)                                    ; ************************************************************************************
1392
1393  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01              7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                CALL    TSET            ;SETUP THE TEST
1394                                    ;%RPM3-READ PATH-IBG DETECT
1395                                    ;&M8953, M8950'S
1396  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28         18.0                        CALL    REQST
 (1)  4308  00                                             .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                         .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                                         .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                             .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                             .BYTE   @7              ;REQUEST CODE
1397  430F                                      RIN     R12L
 (1)  430F  DB  94              10.0                       IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                   4.0                       MOV     A,A             ;RETRY LINK
1398  4312  32  FD  4A          13.0              STA     UNITMP
1399  4315  A7                    4.0             ANA     A
1400  4316  C2  27  43          10.0              JNZ     UNITSL
1401  4319                                      ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28         18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                             .BYTE
 (1)  431E  CD  15  28         18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  CA  48         10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1402                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1403                                    ;>FATAL ERROR - TEST ABORTED
1404  4324  C3  CA  48          10.0              JMP     EXIT
1405
1406  4327  06  00                7.0   UNITSL: MVI     B,0             ;
```

```
1407  4329  3A FD 4A    13.0          LDA   UNITMP          ;DID THE USER SPECIFY TU
1408  432C  E6 01        7.0           ANI   @01             ;PORT 0?
1409  432E  C2 44 43     10.0          JNZ   FOUND           ;YES-GO USE PORT #0
1410  4331  04            4.0          INR   B               ;NO-UPDATE POINTER TO PORT #1
1411  4332  3A FD 4A    13.0          LDA   UNITMP          ;DID THE USER SPECIFY TU
1412  4335  E6 02        7.0           ANI   @02             ;PORT 1?
1413  4337  C2 44 43     10.0          JNZ   FOUND           ;YES-GO USE PORT #1
1414  433A  04            4.0          INR   B               ;NO-UPDATE POINTER TO PORT #2
1415  433B  3A FD 4A    13.0          LDA   UNITMP          ;DID THE USER SPECIFY TU
1416  433E  E6 04        7.0           ANI   @04             ;PORT 2?
1417  4340  C2 44 43     10.0          JNZ   FOUND           ;YES-GO USE PORT #2
1418  4343  04            4.0          INR   B               ;NO-ASSUME PORT #3
1419  4344  CD CD 48    18.0  FOUND:  CALL  CLEAR           ;CLEAL ALL TU PORTS
1420  4347  DB E0        10.0          IN    INTSTA          ;GET THE MASS BUS PORT
1421  4349  E6 80        7.0           ANI   BIT7            ;
1422  434B  B0            4.0          ORA   B               ;OR IN THE TU PORT
1423  434C  D3 E0        10.0          OUT   MBSEL           ;SELECT DESIRED PORT
1424  434E  32 FE 4A    13.0          STA   SUNIT           ;SAVE FOR LATER
1425  4351  3E 10        7.0   TST01X: MVI   A,RDCLK         ;LOAD THE NORMAL READ PATH CLOCK
1426  4353  D3 F0        10.0          OUT   CLKCTL          ;SET NORMAL CLOCKS
1427
1428  4355  3E 80        7.0           MVI   A,R.IBJN                ;LOAD THE READ PATH ENABLE BIT
1429  4357  D3 09        10.0          OUT   RPCTL           ;ENABLE THE READ PATH
1430
1431  4359  21 00 00    10.0          LXI   H,@0            ;INITIALIZE THE AMTIE MASK TO ZERO
1432  435C  22 FF 4A    16.0          SHLD  AMTMSK
1433
1434  435F  3E 04        7.0   AMDLP1: MVI   A,P.RPEN
1435  4361  D3 4C        10.0          OUT   PENAB
1436  4363  3E 21        7.0           MVI   A,P.LWR.@1      ;SET DP ENABLE AND LOOP W/R AND AMTIE P
1437  4365  D3 48        10.0          OUT   PDIAG           ;SET THE PARITY AMTIE
1438  4367  3E FF        7.0           MVI   A,@377          ;SET AMTIES 7:0
1439  4369  D3 44        10.0          OUT   TAMT
1440  436B  06 0A        7.0           MVI   B,10            ;PERFORM A 50 MICROSECOND
1441  436D  05            4.0   2$:     DCR   B               ;TIMEOUT
1442  436E  C2 6D 43     10.0          JNZ   2$
1443
1444  4371  3E 01        7.0           MVI   A,RIBG          ;LOAD THE READ INTERBLOCK GAP CMD.
1445  4373  D3 0B        10.0          OUT   RCMD            ;ISSUE THE COMMAND
1446
1447  4375  06 0A        7.0           MVI   B,10            ;PERFORM A 50 MICROSECOND
1448  4377  05            4.0   1$:     DCR   B               ;TIMEOUT
1449  4378  C2 77 43     10.0          JNZ   1$
1450
1451  437B  DB 02        10.0          IN    RSTAT           ;GET THE READ PATH STATUS
1452  437D  A7            4.0          ANA   A               ;SET CONDITION BITS
1453  437E  CA 8D 43     10.0          JZ    AMDCN1          ;CONTINUE IF ZERO
1454  4381                             ROUT  ADATA           ;SAVE THE ACTUAL STATUS
 (1)  4381  D3 94        10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  4383  7F            4.0                MOV   A,A               ;RETRY LINK
1455  4384  AF            4.0          XRA   A               ;CLEAR THE ACCUMULATOR
1456  4385                             ROUT  EDATA           ;STORE AS EXPECTED STATUS
 (1)  4385  D3 95        10.0                OUT   EDATA             ;WRITE AC INTO EDATA
 (1)  4387  7F            4.0                MOV   A,A               ;RETRY LINK
```

```
1457  4388                                                ERRB    AMDLP1,AMDCN1
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4388  CD  12  28        18.0                                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0002                                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4388  02                                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  438C  00                                                     .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  438D  CD  15  28        18.0                         AMDCN1::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4390  DA  5F  43        10.0                                 JC      AMDLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1458                                                      ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD IBG
1459  4393  3A  00  4B        13.0                 AMDLP2: LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
1460  4396  F6  20              7.0                         ORI     P.LWR           ;SET LOOP W/R
1461  4398  D3  48             10.0                         OUT     PDIAG           ;AS PER THE MASK
1462  439A  3A  FF  4A         13.0                         LDA     AMTMSK          ;SET THE AMTIE BITS 7:0
1463  439D  D3  44             10.0                         OUT     TAMT            ;AS PER THE MASK
1464
1465  439F  06  0A              7.0                         MVI     B,10            ;PERFORM A 50 MICROSECOND
1466  43A1  05                  4.0                 2$:     DCR     B               ;TIMEOUT
1467  43A2  C2  A1  43         10.0                         JNZ     2$
1468
1469  43A5  DB  02             10.0                         IN      RSTAT           ;GET THE READ PATH STATUS
1470  43A7  FE  90              7.0                         CPI     BADST           ;BAD STATUS?
1471  43A9  CA  CE  43         10.0                         JZ      AMDCN2          ;YES-CONTINUE
1472  43AC                                                  ROUT    ADATA           ;NO-STORE ACTUAL STATUS
 (1)  43AC  D3  94             10.0                                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43AE  7F                  4.0                                 MOV     A,A             ;RETRY LINK
1473  43AF  3E  90              7.0                         MVI     A,BADST         ;GET EXPECTED STATUS
1474  43B1                                                  ROUT    EDATA           ;STORE EXPECTED STATUS
 (1)  43B1  D3  95             10.0                                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  43B3  7F                  4.0                                 MOV     A,A             ;RETRY LINK
1475
1476  43B4  3A  FF  4A         13.0                 LDA     AMTMSK          ;GET AMTIE BITS 7:0
1477  43B7                                          ROUT    R16L            ;STORE IN CAS
 (1)  43B7  D3  9C             10.0                         OUT     R16L            ;WRITE AC INTO R16L
 (1)  43B9  7F                  4.0                         MOV     A,A             ;RETRY LINK
1478  43BA  3A  00  4B         13.0                 LDA     AMTMSK+1        ;GET AMTIE PARITY BIT
1479  43BD                                          ROUT    R16H            ;STORE IN CAS
 (1)  43BD  D3  9D             10.0                         OUT     R16H            ;WRITE AC INTO R16H
 (1)  43BF  7F                  4.0                         MOV     A,A             ;RETRY LINK
1480  43C0  3E  80              7.0                 MVI     A,@200          ;SET THE ACTUAL DATA PRINT FLAG
1481  43C2                                          ROUT    R17L            ;STORE IN CAS
 (1)  43C2  D3  9E             10.0                         OUT     R17L            ;WRITE AC INTO R17L
 (1)  43C4  7F                  4.0                         MOV     A,A             ;RETRY LINK
1482  43C5  AF                  4.0                 XRA     A               ;CLEAR THE EXPECTED DATA
1483  43C6                                          ROUT    R15L            ;PRINT FLAG
 (1)  43C6  D3  9A             10.0                         OUT     R15L            ;WRITE AC INTO R15L
 (1)  43C8  7F                  4.0                         MOV     A,A             ;RETRY LINK
```

```
1485  43C9                                                ERRB    AMDLP2,AMDCN2,@5
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  43C9   CD   12   28        18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)         0003                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43CC   03                                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43CD   05                                                  .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  43CE   CD   13   28        18.0                    AMDCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43D1   DA   93   43        10.0                            JC      AMDLP2          ;LOOP ADDRESS IF LOOP SPECIFIED
1486                                                    ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD IBG
1487                                                    ;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1488  43D4   2A   FF   4A        16.0            LHLD    AMTMSK          ;GET THE AMTIE MASK
1489  43D7   23                    6.0            INX     H               ;INCREMENT BY 1
1490  43D8   22   FF   4A        16.0            SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1491  43DB   3A   FF   4A        13.0            LDA     AMTMSK          ;GET LS 8 BITS OF AMTIE MASK
1492  43DE   FE   FF               7.0            CPI     @377            ;ALL ONES?
1493  43E0   C2   5F   43        10.0            JNZ     AMDLP1          ;NO-CONTINUE
1494  43E3   3A   00   4B        13.0            LDA     AMTMSK+1        ;YES-GET AMTIE PARITY BIT
1495  43E6   FE   01               7.0            CPI     @1              ;END OF TEST
1496  43E8   C2   5F   43        10.0            JNZ     AMDLP1          ;NO-CONTINUE
1497  43EB                                        ENDTST  TST01X          ;YES-END IT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43EB                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43EB   CD   06   28        18.0                    CALL    REQST
 (2)  43EE   00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  43EF   00   00                                     .WORD                   ;SYSTEM '*' COUNT
 (2)  43F1   00   00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43F3   00                                          .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  43F4   07                                          .BYTE   7               ;REQUEST CODE
 (1)  43F5   3A   9A   4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  43F8   3D                   4.0            DCR     A               ;DOWNCOUNT
 (1)  43F9   32   9A   4F        13.0            STA     ITERA           ;SAVE COUNT
 (1)  43FC   F2   51   43        10.0            JP      TST01X          ;DO TEST UNTIL TILL = 0
1498  43FF   CD   CD   48        18.0            CALL    CLEAR
1499
1500
```

```
                                .SBTTL  TEST 02 - PE ID BURST TEST

                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE ID BURST TEST
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST IS DESIGNED TO CHECK THE READ PATHS ABILITY TO DETECT A
                                ;*CORRECT PE IDENTIFICATION BURST AND TO REJECT INCORRECT ONES.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     ENABLE THE READ MICRO CONTROLLER
                                ;*     INIT THE AMTIE MASK TO ZERO (9 BITS WIDE)
                                ;*     BGNDO
                                ;*        :    SET UP THE AMTIES FOR A GOOD ID BURST
                                ;*        :    DELAY 50 MICROSECONDS
                                ;*        :    ISSUE THE ID BURST COMMAND
                                ;*        :    DELAY 50 MICROSECONDS
                                ;*        :    READ THE READ MICRO STATUS
                                ;*        :         IF THE STATUS = 0
                                ;*        :         :    THEN-CONTINUE
                                ;*        :         :    ELSE-ERROR
                                ;*        :         ENDIF
                                ;*        :    ISSUE THE AMTIE LINES AS PER THE AMTIE MASK
                                ;*        :    DELAY 50 MICROSECONDS
                                ;*        :    READ THE READ MICRO STATUS
                                ;*        :         IF STATUS = 'BADST''
                                ;*        :         :    THEN-CONTINUE
                                ;*        :         :    ELSE-ERROR
                                ;*        :         ENDIF
                                ;*        :    INCREMENT THE AMTIE MASK BY ONE
                                ;*        :         IF AMTIE MASK - 377(8)
                                ;*        :         :    THEN-INCREMENT THE AMTIE MASK BY ONE
                                ;*        :         :    ELSE-CONTINUE
                                ;*        :         ENDIF
                                ;*        :    DO UNTIL AMTIE MASK   1000(8)
                                ;*     ENDDO
                                ;*ENDTST
                                SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM3 MICRO TEST 02
                                ;*RPM3 MICRO ERROR 04
                                ;*RPM3-READ PATH-PE ID BURST DETECT
```

```
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD PE ID
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM3 MICRO TEST 02
                                        ;*RPM3 MICRO ERROR 05
                                        ;*RPM3-READ PATH-PE ID BURST DETECT
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE ID
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT = AMTIE LINES
4402                                    S
 (1)                                    ; ************************************************************************************

                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
4402                                                    MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4402  3E  02              7.0                     CALL    TSET            ;SETUP THE TEST
 (1)  4404  CD  03  28         18.0
                                        ;%RPM3-READ PATH-PE ID BURST DETECT
                                        ;&M8953, M8950'S
4407  3E  10                    7.0     TST02X: MVI     A,RDCLK         ;LOAD THE NORMAL READ PATH CLOCK
4409  D3  F0                   10.0             OUT     CLKCTL          ;SET NORMAL CLOCKS
440B  CD  CD  48               18.0             CALL    CLEAR           ;CLEAR ALL TU PORTS
440E  3E  80                    7.0             MVI     A,R.TBJN        ;LOAD THE READ PATH ENABLE BIT
4410  D3  09                   10.0             OUT     RPCTL           ;ENABLE THE READ PATH
4412  21  00  00               10.0             LXI     H,@0            ;INITIALIZE THE AMTIE MASK TO ZERO
4415  22  FF  4A               16.0             SHLD    AMTMSK
4418  3A  FE  4A               13.0     AMDLP7: LDA     SUNIT
441B  D3  E0                   10.0             OUT     MBSEL
441D  3E  04                    7.0             MVI     A,P.RPEN
441F  D3  4C                   10.0             OUT     PENAB
4421  3E  20                    7.0             MVI     A,P.LWR         ;CLEAR THE PARITY AMTIE
4423  D3  48                   10.0             OUT     PDIAG           ;SET THE PARITY AMTIE
4425  3E  FF                    7.0             MVI     A,@377          ;SET AMTIES 7:0
4427  D3  44                   10.0             OUT     TAMT
4429  06  0A                    7.0             MVI     B,10            ;PERFORM A 50 MICROSECOND
442B  05                       4.0     2$:     DCR     B               ;TIMEOUT
442C  C2  2B  44               10.0             JNZ     2$
442F  3E  02                    7.0             MVI     A,RPEI          ;LOAD THE READ PE ID BURST CMD.
4431  D3  0B                   10.0             OUT     RCMD            ;ISSUE THE COMMAND
4433  06  0A                    7.0             MVI     B,10            ;PERFORM A 50 MICROSECOND
4435  05                       4.0     1$:     DCR     B               ;TIMEOUT
4436  C2  35  44               10.0             JNZ     1$
4439  DB  02                   10.0             IN      RSTAT           ;GET THE READ PATH STATUS
443B  A7                       4.0              ANA     A               ;SET CONDITION BITS
443C  CA  4B  44               10.0             JZ      AMDCN7          ;CONTINUE IF ZERO
443F                                            ROUT    ADATA           ;SAVE THE ACTUAL STATUS
443F  D3  94                   10.0                             OUT     ADATA           ;WRITE AC INTO ADATA
4441  7F                       4.0                              MOV     A,A             ;RETRY LINK
4442  AF                       4.0              XRA     A               ;CLEAR THE ACCUMULATOR
4443                                            RUUT    EDATA           ;STORE AS EXPECTED STATUS
```

```
                                        OUT     EDATA           ;WRITE AC INTO EDATA
                                        MOV     A,A             ;RETRY LINK
                                ERRB    AMDLP7,AMDCN7
                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
                                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                        .BYTE                   ;PRINT ROUTINE NUMBER
                                        AMDCN7::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      AMDLP7          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD PE ID
                                AMDLP8: LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
                                        ORI     P.LWR           ;SET LOOP W/R
                                        OUT     PDIAG           ;AS PER THE MASK
                                        LDA     AMTMSK          ;SET THE AMTIE BITS 7:0
                                        OUT     TAMT            ;AS PER THE MASK
                                        MVI     B,10            ;PERFORM A 50 MICROSECOND
                                2$:     DCR     B               ;TIMEOUT
                                        JNZ     2$
                                        IN      RSTAT           ;GET THE READ PATH STATUS
                                        CPI     BADST           ;BAD STATUS?
                                        JZ      AMDCN8          ;YES-CONTINUE
                                        ROUT    ADATA           ;NO-STORE ACTUAL STATUS
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        MVI     A,BADST         ;GET EXPECTED STATUS
                                        ROUT    EDATA           ;STORE EXPECTED STATUS
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        LDA     AMTMSK          ;GET AMTIE BITS 7:0
                                        ROUT    R16L            ;STORE IN CAS
                                                OUT     R16L            ;WRITE AC INTO R16L
                                                MOV     A,A             ;RETRY LINK
                                LDA     AMTMSK+1                ;GET AMTIE PARITY BIT
                                ROUT    R16H                    ;STORE IN CAS
                                                OUT     R16H            ;WRITE AC INTO R16H
                                                MOV     A,A             ;RETRY LINK
                                MVI     A,@200                  ;SET THE ACTUAL DATA PRINT FLAG
                                ROUT    R17L                    ;STORE IN CAS
                                                OUT     R17L            ;WRITE AC INTO R17L
                                                MOV     A,A             ;RETRY LINK
                                XRA     A                       ;CLEAR THE EXPECTED DATA
                                ROUT    R15L                    ;PRINT FLAG
                                                OUT     R15L            ;WRITE AC INTO R15L
                                                MOV     A,A             ;RETRY LINK
```

```
1618  4487                                                  ERRB    AMDLP8,AMDCN8,@5
 (1)                                                        ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4487  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448A  05                                             .BYTF   MSGN            ;MESSAGE NUMBER ID
 (1)  448B  05                                             .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  448C  CD  15  28        18.0                AMDCN8::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  448F  DA  51  44        10.0                        JC      AMDLP8          ;LOOP ADDRESS IF LOOP SPECIFIED
1619                                            ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE ID
1620                                            ;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1621  4492  2A  FF  4A        16.0              3$:     LHLD    AMTMSK          ;GET THE AMTIE MASK
1622  4495  23                  6.0                      INX     H               ;INCREMENT
1623  4496  22  FF  4A        16.0                      SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1624  4499  3A  00  4B        13.0                      LDA     AMTMSK+1        ;GET THE AMTIE MASK PARITY BIT
1625  449C  FE  02                7.0                   CPI     @2              ;DONE?
1626  449E  CA  B1  44        10.0                      JZ      AMDLP9          ;YES-END OF TEST
1627  44A1  FE  01                7.0                   CPI     @1              ;RESULT > 377?
1628  44A3  CA  18  44        10.0                      JZ      AMDLP7          ;YES-CONTINUE
1629  44A6  3A  FF  4A        13.0                      LDA     AMTMSK          ;NO-GET THE AMTIE DATA MASK
1630  44A9  FE  FF                7.0                   CPI     @377            ;=377?
1631  44AB  CA  92  44        10.0                      JZ      3$              ;YES-INCREMENT AGAIN
1632  44AE  C3  18  44        10.0                      JMP     AMDLP7          ;NO-CONTINUE
1633
1634  44B1                                          AMDLP9: ENDTST  TST02X
 (1)                                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44B1                                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44B1  CD  06  28        18.0                                          CALL    REQST
 (2)  44B4  00                                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  44B5  00  00                                                          .WORD                   ;SYSTEM '*' COUNT
 (2)  44B7  00  00                                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44B9  00                                                              .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  44BA  07                                                              .BYTE   7               ;REQUEST CODE
 (1)  44BB  3A  9A  4F        13.0                          LDA     ITERA           ;GET ITERATION COUNT
 (1)  44BE  3D                  4.0                         DCR     A               ;DOWNCOUNT
 (1)  44BF  32  9A  4F        13.0                          STA     ITERA           ;SAVE COUNT
 (1)  44C2  F2  07  44        10.0                          JP      TST02X          ;DO TEST UNTIL TILL = 0
1635  44C5  CD  CD  48        18.0                          CALL    CLEAR
1636
1637
```

```
                                .SBTTL  TEST 03 - GCR ID BURST TEST

                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR ID BURST TEST
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST IS DESIGNED TO CHECK THE READ PATHS ABILITY TO DETECT A
                                ;*CORRECT GCR IDENTIFICATION BURST AND TO REJECT INCORRECT ONES.
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     ENABLE THE READ MICRO CONTROLLER
                                ;*     INIT THE AMTIE MASK TO ZERO (9 BITS WIDE)
                                ;*     BGNDO
                                ;*        :    SET UP THE AMTIES FOR A GOOD GCR ID BURST
                                ;*        :    DELAY 50 MICROSECONDS
                                ;*        :    ISSUE THE ID BURST COMMAND
                                ;*        :    DELAY 50 MICROSECONDS
                                ;*        :    READ THE READ MICRO STATUS
                                ;*        :         IF THE STATUS = 0
                                ;*        :         :    THEN-CONTINUE
                                ;*        :         :    ELSE-ERROR
                                ;*        :         ENDIF
                                ;*        :    ISSUE THE AMTIE LINES AS PER THE AMTIE MASK
                                ;*        :    DELAY 50 MICROSECONDS
                                ;*        :    READ THE READ MICRO STATUS
                                ;*        :         IF STATUS = 'BADST''
                                ;*        :         :    THEN-CONTINUE
                                ;*        :         :    ELSE-ERROR
                                ;*        :         ENDIF
                                ;*        :    INCREMENT THE AMTIE MASK BY ONE
                                ;*        :         IF AMTIE MASK = 677(8)
                                ;*        :         :    THEN-INCREMEN' THE AMTIE MASK BY ONE
                                ;*        :         :    ELSE-ERROR
                                ;*        :         ENDIF
                                ;*        :    DO UNTIL AMTIE MASK - 1000(8)
                                ;*     ENDDO
                                ;*ENDTST
                                SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM3 MICRO TEST 03
                                ;*RPM3 MICRO ERROR 06
                                ;*RPM3-READ PATH-GCR ID BURST DETECT
```

```
1681                                    ;*M8953, M8950'S
1682                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD GCR ID
1683                                    ;*ACTUAL = NNNN
1684                                    ;*EXPECTED = NNNN
1685                                    ;*
1686                                    ;*RPM3 MICRO TEST 03
1687                                    ;*RPM3 MICRO ERROR 07
1688                                    ;*RPM3-READ PATH-GCR ID BURST DETECT
1689                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE ID
1690                                    ;*ACTUAL = NNNN
1691                                    ;*EXPECTED = NNNN
1692                                    ;*EXPECTED 18 BITS =    E EEEEEE
1693                                    ;*ACTUAL 18 BITS =      A AAAAAA
1694                                    ;*EXTENDED ACTUAL PRINTOUT = AMIE LINES
1695  44C8                              S
 (1)                                    ; ***************************************************************************
1696
1697  44C8                              TEST3:  TESTX   @3              ;INITIALIZE THE TEST
 (1)  44C8  3E  03          7.0                     MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  44CA  CD  03  28      18.0                    CALL    TSET            ;SETUP THE TEST
1698                                    ;%RPM3-READ PATH-GCR ID BURST DETECT
1699                                    ;&M8953, M8950'S
1700
1701  44CD  3E  10          7.0       TST03X: MVI     A,RDCLK         ;LOAD THE NORMAL READ PATH CLOCK
1702  44CF  D3  F0         10.0               OUT     CLKCTL          ;SET NORMAL CLOCKS
1703  44D1  CD  CD  48      18.0               CALL    CLEAR           ;CLEAR ALL TU PORTS
1704
1705  44D4  3E  80          7.0               MVI     A,R.TBJN        ;LOAD THE READ PATH ENABLE BIT
1706  44D6  D3  09         10.0               OUT     RPCTL           ;ENABLE THE READ PATH
1707  44D8  21  00  00      10.0               LXI     H,@0            ;INITIALIZE THE AMTIE TO ZERO
1708  44DB  22  FF  4A      16.0               SHLD    AMTMSK
1709  44DE  3A  FE  4A      13.0     AMDLPA:   LDA     SUNIT
1710  44E1  D3  E0         10.0               OUT     MBSEL
1711  44E3  3E  04          7.0               MVI     A,P.RPEN
1712  44E5  D3  4C         10.0               OUT     PENAB
1713  44E7  3E  21          7.0               MVI     A,P.LWR.@1      ;SET THE PARITY AMTIE
1714  44E9  D3  48         10.0               OUT     PDIAG           ;SET THE PARITY AMTIE
1715  44EB  3E  BF          7.0               MVI     A,@277          ;SET AMTIES
1716  44ED  D3  44         10.0               OUT     TAMT
1717  44EF  06  0A          7.0               MVI     B,10            ;PERFORM A 50 MICROSECOND
1718  44F1  05              4.0     2$:       DCR     B               ;TIMEOUT
1719  44F2  C2  F1  44      10.0               JNZ     2$
1720  44F5  3E  03          7.0               MVI     A,RGCRI         ;LOAD THE READ GCR ID BURST CMD.
1721  44F7  D3  08         10.0               OUT     RCMD            ;ISSUE THE COMMAND
1722
1723  44F9  06  0A          7.0               MVI     B,10            ;PERFORM A 50 MICROSECOND
1724  44FB  05              4.0     1$:       DCR     B               ;TIMEOUT
1725  44FC  C2  FB  44      10.0               JNZ     1$
1726
1727  44FF  DB  02         10.0               IN      RSTAT           ;GET THE READ PATH STATUS
1728  4501  A7              4.0               ANA     A               ;SET CONDITION BITS
1729  4502  CA  11  45      10.0               JZ      AMDCNA          ;CONTINUE IF ZERO
1730  4505                                    ROUT    ADATA           ;SAVE THE ACTUAL STATUS
 (1)  4505  D3  94         10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
```

```
  (1)  4507  7F                    4.0                     MOV     A,A             ;RETRY LINK
1731   4508  AF                    4.0             XRA     A               ;CLEAR THE ACCUMULATOR
1732   4509                                        ROUT    EDATA           ;STORE AS EXPECTED STATUS
  (1)  4509  D3  95                10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4508  7F                    4.0                     MOV     A,A             ;RETRY LINK
1733   450C                                        ERRB    AMDLPA,AMDCNA
  (1)                                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
  (1)  450C  CD  12  28            18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        0006                                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  450F  06                                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4510  00                                                    .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4511  CD  15  28            18.0            AMDCNA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4514  DA  DE  44            10.0                            JC      AMDLPA          ;LOOP ADDRESS IF LOOP SPECIFIED
1734                                            ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD GCR ID
1735
1736   4517  3A  00  4B            13.0    AMDLPB: LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
1737   451A  F6  20                  7.0             ORI     P.LWR           ;SET LOOP W/R
1738   451C  D3  48                 10.0             OUT     PDIAG           ;AS PER THE MASK
1739   451E  3A  FF  4A              13.0             LDA     AMTMSK          ;SET THE AMTIE BITS 7:0
1740   4521  D3  44                 10.0             OUT     TAMT            ;AS PER THE MASK
1741   4523  06  0A                  7.0             MVI     B,10            ;PERFORM A 50 MICROSECOND
1742   4525  05                     4.0    2$:      DCR     B               ;TIMEOUT
1743   4526  C2  25  45            10.0             JNZ     2$
1744   4529  DB  02                 10.0             IN      RSTAT           ;GET THE READ PATH STATUS
1745   452B  FE  90                  7.0             CPI     BADST           ;BAD STATUS?
1746   452D  CA  52  45            10.0             JZ      AMDCNB          ;YES-CONTINUE
1747   4530                                        ROUT    ADATA           ;NO-STORE ACTUAL STATUS
  (1)  4530  D3  94                10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  4532  7F                    4.0                     MOV     A,A             ;RETRY LINK
1748   4533  3E  90                  7.0             MVI     A,BADST         ;GET EXPECTED STATUS
1749   4535                                        ROUT    EDATA           ;STORE EXPECTED STATUS
  (1)  4535  D3  95                10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4537  7F                    4.0                     MOV     A,A             ;RETRY LINK
1750
1751   4538  3A  FF  4A              13.0             LDA     AMTMSK          ;GET AMTIE BITS 7:0
1752   453B                                        ROUT    R16L            ;STORE IN CAS
  (1)  453B  D3  9C                10.0                    OUT     R16L            ;WRITE AC INTO R16L
  (1)  453D  7F                    4.0                     MOV     A,A             ;RETRY LINK
1753   453E  3A  00  4B              13.0             LDA     AMTMSK+1        ;GET AMTIE PARITY BIT
1754   4541                                        ROUT    R16H            ;STORE IN CAS
  (1)  4541  D3  9D                10.0                    OUT     R16H            ;WRITE AC INTO R16H
  (1)  4543  7F                    4.0                     MOV     A,A             ;RETRY LINK
1755   4544  3E  80                  7.0             MVI     A,@200          ;SET THE ACTUAL DATA PRINT FLAG
1756   4546                                        ROUT    R17L            ;STORE IN CAS
  (1)  4546  D3  9E                10.0                    OUT     R17L            ;WRITE AC INTO R17L
  (1)  4548  7F                    4.0                     MOV     A,A             ;RETRY LINK
1757   4549  AF                    4.0             XRA     A               ;CLEAR THE EXPECTED DATA
1758   454A                                        ROUT    R15L            ;PRINT FLAG
  (1)  454A  D3  9A                10.0                    OUT     R15L            ;WRITE AC INTO R15L
  (1)  454C  7F                    4.0                     MOV     A,A             ;RETRY LINK
```

```
                                        ERRB    AMDLPB,AMDCNB,@5
                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE   @5              ;PRINT ROUTINE NUMBER
                                        AMDCNB::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                JC      AMDLPB          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE TD
                                ;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1763 4558 2A FF 4A      16.0    3$:     LHLD    AMTMSK          ;GET THE AMTIE MASK
1764 455B 23             6.0            INX     H               ;INCREMENT
1765 455C 22 FF 4A      16.0            SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1766 455F 3A 00 4B      13.0            LDA     AMTMSK+1        ;GET THE PARITY AMTIE MASK
1767 4562 FE 02          7.0            CPI     @2              ;DONE?
1768 4564 CA 77 45      10.0            JZ      AMDLPC          ;YES-EXIT THE TEST
1769 4567 FE 01          7.0            CPI     @1              ;NO-MASK=677?
1770 4569 C2 DE 44      10.0            JNZ     AMDLPA          ;NO-CONTINUE
1771 456C 3A FF 4A      13.0            LDA     AMTMSK          ;MAYBE
1772 456F FE BF          7.0            CPI     @277            ;LO MASK=277?
1773 457' CA 58 45      10.0            JZ      3$              ;YES-INCREMENT MASK AGAIN
1774 4574 C3 DE 44      10.0            JMP     AMDLPA          ;NO-CONTINUE
1775
1776 4577                               AMDLPC: ENDTST  TST03X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
                                                CALL    REQST
                                                .BYTE                   ;DATA PATTERN NUMBER
                                                .WORD                   ;SYSTEM '*' COUNT
                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                .BYTE                   ;DATA COMPARE FLAG IF =1
                                                .BYTE   7               ;REQUEST CODE
1777 4581 3A 9A 4F      13.0            LDA     ITERA           ;GET ITERATION COUNT
1778 4584 3D             4.0            DCR     A               ;DOWNCOUNT
     4585 32 9A 4F      13.0            STA     ITERA           ;SAVE COUNT
     4588 F2 CD 44      10.0            JP      TST03X          ;DO TEST UNTIL TILL = 0
     458B CD CD 48      18.0            CALL    CLEAR
```

```
                                .SBTTL  TEST 04 - GCR ARA BURST TEST

                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR ARA BURST TEST
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE READ PATHS ABILITY TO DETECT A CORRECT GCR ARA
                                ;*ID BURST, AND REJECT INCORRECT ONES.
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     ENABLE THE READ MICRO CONTROLELR
                                ;*     INIT THE AMTIE MASK TO 001(8) (9 BITS WIDE)
                                ;*       BGNDO
                                ;*         :   CLEAR ALL AMTIES FOR A GOOD ARA BURST
                                ;*         :   DELAY 50 MICROSECONDS
                                ;*         :   ISSUE THE ARA BURST COMMAND
                                ;*         :   DELAY 50 MICRO SECONDS
                                ;*         :   READ THE READ MICRO STATUS
                                ;*         :       IF STATUS = 0
                                ;*         :         :   THEN-CONTINUE
                                ;*         :         :   ELSE-ERROR
                                ;*         :       ENDIF
                                ;*         :   ISSUE THE AMTIE LINES AS PER THE AMTIE MASK
                                ;*         :   DELAY 50 MICRO SECONDS
                                ;*         :   READ THE READ MICRO STATUS
                                ;*         :       IF STATUS = 'BADST''
                                ;*         :         :   THEN-CONTINUE
                                ;*         :         :   ELSE-ERROR
                                ;*         :       ENDIF
                                ;*         :   INCREMENT THE AMTIE MASK BY ONE
                                ;*         :   DO UNTIL THE AMTIE MASK - 1000(8)
                                ;*     ENDDO
                                ;*ENDTST
                                SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM3 MICRO TEST 04
                                ;*RPM3 MICRO ERROR 10
                                ;*RPM3-READ PATH-GCR ARA BURST DETECT TEST
                                ;*M8953, M8950'S
                                ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD ARA BURST
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*RPM3 MICRO TEST 04
                                        ;*RPM3 MICRO ERROR 11
                                        ;*RPM3-READ PATH-GCR ARA BURST DETECT TEST
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA BURST
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT=AMTIE LINES
458E                                    S
 (1)                                    ; ************************************************************************************
                                        TEST4:  TESTX   @4              ;INITIALIZE THE TEST
 (1) 458E  3E  04          7.0                   MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1) 4590  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%RPM3-READ PATH-GCR ARA BURST DETECT TEST
                                        ;&M8953, M8950'S

 4593  3E  10               7.0  TST04X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
 4595  D3  F0              10.0           OUT     CLKCTL
 4597  CD  CD  48          18.0           CALL    CLEAR           ;CLEAR ALL TU PORTS

 459A  3E  80               7.0           MVI     A,R.TBJN                ;ENABLE THE READ MICRO CONTROLLER
 459C  D3  09              10.0           OUT     RPCTL

 459E  21  01  00          10.0           LXI     H,@1            ;INIT THE AMTIE MASK TO 1
 45A1  22  FF  4A          16.0           SHLD    AMTMSK

 45A4  3A  FE  4A          13.0  AMDLPD:  LDA     SUNIT
 45A7  D3  E0              10.0           OUT     MBSEL
 45A9  3E  04               7.0           MVI     A,P.RPEN
 45AB  D3  4C              10.0           OUT     PENAB
 45AD  3E  20               7.0           MVI     A,P.LWR         ;CLEAR THE PARITY AMTIE
 45AF  D3  48              10.0           OUT     PDIAG
 45B1  AF                   4.0           XRA     A               ;CLEAR ALL AMTIES
 45B2  D3  44              10.0           OUT     TAMT
 45B4  06  0A               7.0           MVI     B,10            ;PERFORM A 50 MICROSECOND
 45B6  05                   4.0  2$:      DCR     B               ;TIMEOUT
 45B7  C2  B6  45          10.0           JNZ     2$

 45BA  3E  06               7.0           MVI     A,RARA          ;LOAD THE ARA BURST COMMAND
 45BC  D3  0B              10.0           OUT     RCMD            ;ISSUE THE COMMAND
 45BE  06  0A               7.0           MVI     B,10            ;PERFORM A 50 MICROSECOND
 45C0  05                   4.0  1$:      DCR     B               ;TIMEOUT
 45C1  C2  C0  45          10.0           JNZ     1$

 45C4  DB  02              10.0           IN      RSTAT           ;GET THE READ PATH STATUS
 45C6  A7                   4.0           ANA     A               ;SET CONDITION BITS
 45C7  CA  D6  45          10.0           JZ      AMDCND          ;CONTINUE IF ZERO
                                          ROUT    ADATA           ;SAVE THE ACTUAL STATUS
 45CA
 (1) 45CA  D3  94          10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 45CC  7F               4.0                   MOV     A,A             ;RETRY LINK
```

```
1871  45CD  AF                    4.0          XRA     A                ;CLEAR THE ACCUMULATOR
1872  45CE                                     ROUT    EDATA            ;STORE AS EXPECTED STATUS
 (1)  45CE  D3  95                10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  45D0  7F                    4.0                  MOV     A,A              ;RETRY LINK
1873  45D1                                     ERRB    AMDLPD,AMDCND
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  45D1  CD  12  28            18.0                 CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0008                               MSGN    =       MSGN+1                   ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D4  08                               .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  45D5  00                               .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  45D6  CD  15  28            18.0       AMDCND::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45D9  DA  A4  45            10.0                 JC      AMDLPD           ;LOOP ADDRESS IF LOOP SPECIFIED
1874                                     ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD ARA BURST
1875
1876  45DC  3A  00  4B            13.0  AMDLPE: LDA     AMTMSK+1         ;SET THE AMTIE PARITY BIT
1877  45DF  F6  20                 7.0          ORI     P.LWR            ;SET LOOP W/R
1878  45E1  D3  48                10.0          OUT     PDIAG            ;AS PER THE MASK
1879  45E3  3A  FF  4A            13.0          LDA     AMTMSK           ;SET AMTIE BITS 7:0
1880  45E6  D3  44                10.0          OUT     TAMT             ;AS PER THE MASK
1881
1882  45E8  06  0A                 7.0          MVI     B,10             ;PERFORM A 50 MICRO SECOND
1883  45EA  05                     4.0  2$:     DCR     B                ;TIMEOUT
1884  45EB  C2  EA  45            10.0          JNZ     2$
1885  45EE  DB  02                10.0          IN      RSTAT            ;GET THE READ PATH STATUS
1886  45F0  FE  90                 7.0          CPI     BADST            ;BAD STATUS
1887  45F2  CA  17  46            10.0          JZ      AMDCNE           ;YES-CONTINUE
1888  45F5                                     ROUT    ADATA            ;NO-STORE ACTUAL STATUS
 (1)  45F5  D3  94                10.0                 OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  45F7  7F                    4.0                  MOV     A,A              ;RETRY LINK
1889  45F8  3E  90                 7.0          MVI     A,BADST          ;GET EXPECTED STATUS
1890  45FA                                     ROUT    EDATA            ;STORE EXPECTED STATUS
 (1)  45FA  D3  95                10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  45FC  7F                    4.0                  MOV     A,A              ;RETRY LINK
1891  45FD  3A  FF  4A            13.0          LDA     AMTMSK           ;GET THE AMTIE BITS 7:0
1892  4600                                     ROUT    R16L             ;STORE IN CAS
 (1)  4600  D3  9C                10.0                 OUT     R16L             ;WRITE AC INTO R16L
 (1)  4602  7F                    4.0                  MOV     A,A              ;RETRY LINK
1893  4603  3A  00  4B            13.0          LDA     AMTMSK+1         ;GET AMTIE PARITY
1894  4606                                     ROUT    R16H             ;STORE IN CAS
 (1)  4606  D3  9D                10.0                 OUT     R16H             ;WRITE AC INTO R16H
 (1)  4608  7F                    4.0                  MOV     A,A              ;RETRY LINK
1895  4609  3E  80                 7.0          MVI     A,@200           ;SET ACTUAL DATA PRINT FALG
1896  460B                                     ROUT    R17L             ;CLEAR THE EXPECTED DATA PRINT
 (1)  460B  D3  9E                10.0                 OUT     R17L             ;WRITE AC INTO R17L
 (1)  460D  7F                    4.0                  MOV     A,A              ;RETRY LINK
1897  460E  AF                     4.0          XRA     A                ;FLAG
1898  460F                                     ROUT    R15L
 (1)  460F  D3  9A                10.0                 OUT     R15L             ;WRITE AC INTO R15L
 (1)  4611  7F                    4.0                  MOV     A,A              ;RETRY LINK
```

```
1900  4612                                          ERRB    AMDLPE,AMDCNE,@5
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4612  CD  12  28      18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0009                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4615  09                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4616  05                                          .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  4617  CD  15  28      18.0                AMDCNE::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  461A  DA  DC  45      10.0                        JC      AMDLPE          ;LOOP ADDRESS IF LOOP SPECIFIED
1901                                    ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA BURST
1902                                    ;<EXTENDED ACTUAL PRINTOUT=AMTIE LINES
1903  461D  2A  FF  4A      16.0            LHLD    AMTMSK          ;GET THE AMTIE MASK
1904  4620  23                6.0            INX     H               ;INCREMENT THE MASK
1905  4621  22  FF  4A      16.0            SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1906  4624  3A  00  4B      13.0            LDA     AMTMSK+1        ;GET AMTIE PARITY MASK
1907  4627  FE  02             7.0            CPI     @2              ;DONE?
1908  4629  C2  A4  45      10.0            JNZ     AMDLPD          ;NO-CONTINUE
1909  462C                                  ENDTST  TST04X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  462C                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  462C  CD  06  28      18.0                        CALL    REQST
 (2)  462F  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  4630  00  00                                      .WORD                   ;SYSTEM '*' COUNT
 (2)  4632  C0  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4634  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4635  07                                          .BYTE   7               ;REQUEST CODE
 (1)  4636  3A  9A  4F      13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  4639  3D                4.0            DCR     A               ;DOWNCOUNT
 (1)  463A  32  9A  4F      13.0            STA     ITERA           ;SAVE COUNT
 (1)  463D  F2  93  45      10.0            JP      TST04X          ;DO TEST UNTIL TILL - 0
1910  4640  CD  CD  48      18.0            CALL    CLEAR
```

```
                                .SBTTL  TEST 05 - GCR ARA ID CHARACTER TEST
1913  4643                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*----- -----
                                ;*GCR ARA ID CHARACTER TEST
1915  4643                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE READ PATHS ABILITY TO DETECT A CORRECT GCR ARA
                                ;*ID CHARACTERS, AND REJECT INCORRECT ONES.
1918  4643                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     ENABLE THE READ MICRO CONTROLLER
                                ;*     INIT THE AMTIE MASK TO ZERO (9 BITS)
                                ;*     BGNDO
                                ;*        :    SET UP THE AMTIES FOR A GOOD ARA BURST
                                ;*        :    DELAY 50 MICROSECONDS
                                ;*        :    ISSUE THE ARA ID CHARACTER COMMAND
                                ;*        :    DELAY 50 MICRO SECONDS
                                ;*        :    READ THE READ MICRO STATUS
                                ;*        :         IF THE STATUS = 0
                                ;*        :         :    THEN-CONTINUE
                                ;*        :         :    ELSE-ERROR
                                ;*        :         ENDIF
                                ;*        :    ISSUE THE AMTIE LINES AS PER THE AMTIE MASK
                                ;*        :    DELAY 50 MICRO SECONDS
                                ;*        :    READ THE READ MICRO STATUS
                                ;*        :         IF STATUS = 'BADST''
                                ;*        :         :    THEN-CONTINUE
                                ;*        :         :    ELSE-ERROR
                                ;*        :         ENDIF
                                ;*        :    INCREMENT THE AMTIE MASK BY 1
                                ;*        :         IF AMTIE MASK = 604(8)
                                ;*        :         :    THEN-INCREMENT THE AMTIE MASK BY 1
                                ;*        :         :    ELSE-CONTINUE
                                ;*        :         ENDIF
                                ;*        :    DO UNTIL THE AMTIE MASK - 1000(8)
                                ;*     ENDDO
                                ;*ENDTST
1949  4643                      SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM3 MICRO TEST 05
                                ;*RPM3 MICRO ERROR 12
                                ;*RPM3-READ PATH-GCR ARA ID CHARACTER DETECT
                                ;*M8953, M8950'S
```

```
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD ARA ID CHARACTER
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM3 MICRO TEST 05
                                        ;*RPM3 MICRO ERROR 13
                                        ;*RPM3-READ PATH-GCR ARA ID CHARACTER DETECT
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA ID CHARACTER
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT=AMTIE LINES
                                        $
4643                                    ; ************************************************************************************
 (1)                                    TEST5:  TESTX   @5              ;INITIALIZE THE TEST
 (1) 4643  3E  05            7.0                 MVI     A,@5            ;DEFINE THE TEST NUMBER
 (1) 4645  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%RPM3-READ PATH-GCR ARA ID CHARACTER DETECT
                                        ;&M8953, M8950'S

4648  3E  10                7.0  TST05X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
464A  D3  F0               10.0          OUT     CLKCTL
464C  CD  CD  48           18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS

464F  3E  80                7.0          MVI     A,R.TBJN                ;ENABLE THE READ MICRO CONTROLER
4651  D3  09               10.0          OUT     RPCTL

4653  21  00  00           10.0          LXI     H,@0            ;CLEAR THE AMTIE MASK
4656  22  FF  4A           16.0          SHLD    AMTMSK

4659  3A  FE  4A           13.0  AMDLPG: LDA     SUNIT
465C  D3  E0               10.0          OUT     MBSEL
465E  3E  04                7.0          MVI     A,P.RPEN
4660  D3  4C               10.0          OUT     PENAB
4662  3E  21                7.0          MVI     A,P.LWR!P.AMTP  ;CLEAR THE PARITY AMTIE
4664  D3  48               10.0          OUT     PDIAG
4666  3E  84                7.0          MVI     A,@204          ;SET UP AMTIES 7:0
4668  D3  44               10.0          OUT     TAMT
466A  06  0A                7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
466C  05                    4.0  2$:     DCR     B               ;TIMEOUT
466D  C2  6C  46           10.0          JNZ     2$

4670  3E  04                7.0          MVI     A,RARAI         ;LOAD THE ARA ID CHARACTER COMMAND
4672  D3  08               10.0          OUT     RCMD            ;ISSUE THE COMMAND

4674  06  0A                7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
4676  05                    4.0  1$:     DCR     B               ;TIMEOUT
4677  C2  76  46           10.0          JNZ     1$

467A  DB  02               10.0          IN      RSTAT           ;GET THE READ PATH STATUS
467C  A7                    4.0          ANA     A               ;SET CONDITION BITS
467D  CA  8C  46           10.0          JZ      AMDCNG          ;CONTINUE IF ZERO
```

```
2005  4680                                          ROUT    ADATA           ;SAVE THE ACTUAL STATUS
 (1)  4680  D3  94          10.0                             OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4682  7F               4.0                             MOV     A,A             ;RETRY LINK
2006  4683  AF               4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
2007  4684                                          ROUT    EDATA           ;STORE AS EXPECTED STATUS
 (1)  4684  D3  95          10.0                             OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4686  7F               4.0                             MOV     A,A             ;RETRY LINK
2008  4687                                          ERRB    AMDLPG,AMDCNG
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4687  CD  12  28      18.0                             CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000A                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  468A  0A                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  468B  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  468C  CD  15  28      18.0                 AMDCNG::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  468F  DA  59  46      10.0                             JC      AMDLPG          ;LOOP ADDRESS IF LOOP SPECIFIED
2009                                    ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD ARA ID CHARACTER
2010
2011  4692  3A  00  4B       13.0  AMDLPH: LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
2012  4695  F6  20            7.0          ORI     P.LWR           ;SET LOOP W/R
2013  4697  D3  48           10.0          OUT     PDIAG           ;AS PER THE MASK
2014  4699  3A  FF  4A       13.0          LDA     AMTMSK          ;SET AMTIE BITS 7:0
2015  469C  D3  44           10.0          OUT     TAMT            ;AS PER THE MASK
2016
2017  469E  06  0A            7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
2018  46A0  05                4.0  2$:     DCR     B               ;TIMEOUT
2019  46A1  C2  A0  46       10.0          JNZ     2$
2020
2021  46A4  DB  02           10.0          IN      RSTAT           ;GET THE READ PATH STATUS
2022  46A6  FE  90            7.0          CPI     BADST           ;BAD STATUS
2023  46A8  CA  CD  46       10.0          JZ      AMDCNH          ;YES-CONTINUE
2024  46AB                                  ROUT    ADATA           ;NO-STORE ACTUAL STATUS
 (1)  46AB  D3  94           10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46AD  7F                4.0                     MOV     A,A             ;RETRY LINK
2025  46AE  3E  90            7.0          MVI     A,BADST         ;GET EXPECTED STATUS
2026  46B0                                  ROUT    EDATA           ;STORE EXPECTED STATUS
 (1)  46B0  D3  95           10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46B2  7F                4.0                     MOV     A,A             ;RETRY LINK
2027  46B3  3A  FF  4A       13.0          LDA     AMTMSK          ;GET THE AMTIE BITS 7:0
2028  46B6                                  ROUT    R16L            ;STORE IN CAS
 (1)  46B6  D3  9C           10.0                     OUT     R16L            ;WRITE AC INTO R16L
 (1)  46B8  7F                4.0                     MOV     A,A             ;RETRY LINK
2029  46B9  3A  00  4B       13.0          LDA     AMTMSK+1        ;GET AMTIE PARITY
2030  46BC                                  ROUT    R16H            ;STORE IN CAS
 (1)  46BC  D3  9D           10.0                     OUT     R16H            ;WRITE AC INTO R16H
 (1)  46BE  7F                4.0                     MOV     A,A             ;RETRY LINK
2031  46BF  3E  80            7.0          MVI     A,@200          ;SET ACTUAL DATA PRINT FLAG
2032  46C1                                  ROUT    R17L            ;CLEAR THE EXPECTED DATA PRINT
 (1)  46C1  D3  9E           10.0                     OUT     R17L            ;WRITE AC INTO R17L
 (1)  46C3  7F                4.0                     MOV     A,A             ;RETRY LINK
2033  46C4  AF                4.0          XRA     A               ;FLAG
2034  46C5                                  ROUT    R15L
 (1)  46C5  D3  9A           10.0                     OUT     R15L            ;WRITE AC INTO R15L
 (1)  46C7  7F                4.0                     MOV     A,A             ;RETRY LINK
```

```
2036  46C8                                        ERRB    AMDLPH,AMDCNH,@5
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46C8  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46CB  08                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46CC  05                                            .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  46CD  CD  15  28        18.0                AMDCNH::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46D0  DA  92  46        10.0                        JC      AMDLPH          ;LOOP ADDRESS IF LOOP SPECIFIED
2037                                        ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA ID CHARACTER
2038                                        ;<EXTENDED ACTUAL PRINTOUT=AMTIE LINES
2039
2040  46D3  2A  FF  4A        16.0  1$:   LHLD    AMTMSK          ;GET THE AMTIE MASK
2041  46D6  23                 6.0        INX     H               ;UPDATE THE COUNTER
2042  46D7  22  FF  4A        16.0        SHLD    AMTMSK          ;SAVE IT
2043  46DA  3A  00  4B        13.0        LDA     AMTMSK+1        ;GET AMTIE MASK PARITY BIT
2044  46DD  FE  01             7.0        CPI     @01             ;IS IT 1?
2045  46DF  C2  EA  46        10.0        JNZ     2$              ;NO-CHECK FOR DONE
2046  46E2  3A  FF  4A        13.0        LDA     AMTMSK          ;YES - GET THE LOW ORDER 8 BITS
2047  46E5  FE  84             7.0        CPI     @204            ;DOES THE MASK = 604(8)
2048  46E7  CA  D3  46        10.0        JZ      1$              ;YES-INCREMENT AGAIN
2049  46EA  3A  00  4B        13.0  2$:   LDA     AMTMSK+1        ;GET THE AMTIE PARITY MASK
2050  46ED  FE  02             7.0        CPI     @02             ;DONE?
2051  46EF  C2  59  46        10.0        JNZ     AMDLPG          ;NO-CONTINUE
2052  46F2                                ENDTST  TST05X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46F2                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46F2  CD  06  28        18.0                        CALL    REQST
 (2)  46F5  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  46F6  00  00                                        .WORD                   ;SYSTEM '*' COUNT
 (2)  46F8  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46FA  00                                            .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  46FB  07                                            .BYTE   7               ;REQUEST CODE
 (1)  46FC  3A  9A  4F        13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  46FF  3D                 4.0        DCR     A               ;DOWNCOUNT
 (1)  4700  32  9A  4F        13.0        STA     ITERA           ;SAVE COUNT
 (1)  4703  F2  48  46        10.0        JP      TST05X          ;DO TEST UNTIL TILL - 0
2053  4706  CD  CD  48        18.0        CALL    CLEAR           ;CLEAN UP PORTS
2054
2055
```

```
                                .SBTTL  TEST 06 - TAPE MARK TEST

                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*TAPE MARK TEST
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE READ PATHS ABILITY TO DETECT A TAPE MARK, AND
                                ;*PROPERLY REJECT INCORRECT ONES.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    ENABLE THE READ MICRO CONTROLLER
                                ;*    INITIALIZE THE AMTIE MASK TO ZERO (9 BITS WIDE)
                                ;*    BGNDO
                                ;*       :    SET THE AMTIE LINES TO THE VALUE OF THE AMTIE MASK
                                ;*       :    DELAY 50 MICROSECONDS
                                ;*       :    ISSUE THE TAPE MARK COMMAND
                                ;*       :    DELAY 300 MICRO SECONDS
                                ;*       :    READ THE MICRO STATUS
                                ;*       :       IF STATUS = 0
                                ;*       :       :   THEN-CONTINUE
                                ;*       :       :   ELSE-ERROR
                                ;*       :       ENDIF
                                ;*       :    SET THE AMTIE LINES TO 130(8) FOR A GOOD TAPE MARK
                                ;*       :    DELAY 300 MICRO SECONDS
                                ;*       :    READ THE READ MICRO STATUS
                                ;*       :       IF STATUS-''GOODTM''
                                ;*       :       :   THEN-CONTINUE
                                ;*       :       :   ELSE-ERROR
                                ;*       :       ENDIF
                                ;*       :    INCREMENT THE AMTIE MASK BY 1
                                ;*       :       IF AMTIE MASK-130(8)
                                ;*       :       :   THEN-INCREMENT THE AMTIE MASK BY 1
                                ;*       :       :   ELSE-CONTINUE
                                ;*       :       ENDIF
                                ;*       :    DO UNTIL THE AMTIE MASK   1000(8)
                                ;*    ENDDO
                                ;*ENDTST
                                SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM3 MICRO TEST 06
                                ;*RPM3 MICRO ERROR 14
                                ;*RPM3-READ PATH-TAPE MARK DETECT TEST
```

```
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD TAPE MARK
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT = AMTIE LINES
                                        ;*
                                        ;*RPM3 MICRO TEST 06
                                        ;*RPM3 MICRO ERROR 15
                                        ;*RPM3-READ PATH-TAPE MARK DETECT TEST
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD TAPE MARK
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
2114  4709                              S
 (1)                                    ; ***************************************************************************
2115  4709                              TEST6:  TESTX   @6              ;INITIALIZE THE TEST
 (1)  4709  3E  06              7.0                     MVI     A,@6            ;DEFINE THE TEST NUMBER
 (1)  4708  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
2116                                    ;%RPM3-READ PATH-TAPE MARK DETECT TEST
2117                                    ;&M8953, M8950'S
2118
2119  470E  3E  10               7.0  TST06X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
2120  4710  D3  F0              10.0           OUT     CLKCTL
2121  4712  CD  CD  48          18.0           CALL    CLEAR           ;CLEAR ALL TU PORTS
2122  4715  D3  09              10.0           OUT     RPCTL
2123  4717  21  00  00          10.0           LXI     H,@0            ;CLEAR THE AMTIE MASK
2124  471A  22  FF  4A          16.0           SHLD    AMTMSK
2125  471D  3A  FE  4A          13.0  AMDLPI:  LDA     SUNIT
2126  4720  D3  E0              10.0           OUT     MBSEL
2127  4722  3L  04               7.0           MVI     A,P.RPEN
2128  4724  D3  4C              10.0           OUT     PENAB
2129  4726  3A  00  4B          13.0           LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
2130  4729  F6  20               7.0           ORI     P.LWR           ;SET LOOP W/R
2131  472B  D3  48              10.0           OUT     PDIAG           ;AS PER THE AMTIE MASK
2132  472D  3A  FF  4A          13.0           LDA     AMTMSK          ;SET AMTIE BITS 7:0
2133  4730  D3  44              10.0           OUT     TAMT            ;AS PER THE AMTIE MASK
2134  4732  06  0A               7.0           MVI     B,10            ;PERFORM A 50 MICROSECOND
2135  4734  05                   4.0  2$:      DCR     B               ;TIMEOUT
2136  4735  C2  34  47          10.0           JNZ     2$
2137  4738  3E  05               7.0           MVI     A,RTM           ;LOAD THE TEST TAPE MARK COMMAND
2138  473A  D3  08              10.0           OUT     RCMD            ;ISSUE THE COMMAND
2139
2140  473C  06  3C               7.0           MVI     B,60            ;PERFORM A 300 MICRO SECOND
2141  473E  05                   4.0  1$:      DCR     B               ;TIMEOUT
2142  473F  C2  3E  47          10.0           JNZ     1$
2143
2144  4742  DB  02              10.0           IN      RSTAT           ;GET THE READ PATH STATUS
2145  4744  A7                   4.0           ANA     A               ;SET CONDITION BITS
2146  4745  CA  69  47          10.0           JZ      AMDCNI          ;CONTINUE IF ZERO
2147  4748                               ROUT    ADATA
 (1)  4748  D3  94              10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  474A  7F                   4.0                   MOV     A,A             ;RETRY LINK
```

```
2148  474B  AF                          4.0
2149  474C
 (1)  474C  D3   95                     10.0
 (1)  474E  7F                          4.0
2150  474F  3A   FF   4A                13.0
2151  4752
 (1)  4752  D3   9C                     10.0
 (1)  4754  7F                          4.0
2152  4755  3A   00   4B                13.0
2153  4758
 (1)  4758  D3   9D                     10.0
 (1)  475A  7F                          4.0
2154  475B  3E   80                     7.0
2155  475D
 (1)  475D  D3   9E                     10.0
 (1)  475F  7F                          4.0
2156  4760  AF                          4.0
2157  4761
 (1)  4761  D3   9A                     10.0
 (1)  4763  7F                          4.0
2158  4764
 (1)
 (1)  4764  CD   12   28                18.0
 (1)        000C
 (1)  4767  0C
 (1)  4768  05
 (1)  4769  CD   15   28                18.0
 (1)  476C  DA   1D   47                10.0
2159
2160
2161  476F  3E   20                     7.0
2162  4771  D3   48                     10.0
2163  4773  3E   58                     7.0
2164  4775  D3   44                     10.0
2165  4777  06   3C                     7.0
2166  4779  05                          4.0
2167  477A  C2   79   47                10.0
2168  477D  DB   02                     10.0
2169  477F  FE   92                     7.0
2170  4781  CA   91   47                10.0
2171  4784
 (1)  4784  D3   94                     10.0
 (1)  4786  7F                          4.0
2172  4787  3E   92                     7.0
2173  4789
 (1)  4789  D3   95                     10.0
 (1)  478B  7F                          4.0
```

```
        XRA     A
        ROUT    EDATA
                OUT     EDATA           ;WRITE AC INTO EDATA
                MOV     A,A             ;RETRY LINK
        LDA     AMTMSK
        ROUT    R16L
                OUT     R16L            ;WRITE AC INTO R16L
                MOV     A,A             ;RETRY LINK
        LDA     AMTMSK+1
        ROUT    R16H
                OUT     R16H            ;WRITE AC INTO R16H
                MOV     A,A             ;RETRY LINK
        MVI     A,@200
        ROUT    R17L
                OUT     R17L            ;WRITE AC INTO R17L
                MOV     A,A             ;RETRY LINK
        XRA     A
        ROUT    R15L
                OUT     R15L            ;WRITE AC INTO R15L
                MOV     A,A             ;RETRY LINK
        ERRB    AMDLPI,AMDCNI,@5
        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                        .BYTE   @5              ;PRINT ROUTINE NUMBER
                AMDCNI::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                        JC      AMDLPI          ;LOOP ADDRESS IF LOOP SPECIFIED
;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD TAPE MARK
;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
AMDLPJ: MVI     A,P.LWR         ;CLEAR THE AMTIE PARITY
        OUT     PDIAG
        MVI     A, @130         ;SET AMTIES FOR TRACKS 3, 6 AND 9
        OUT     TAMT
        MVI     B,60            ;PRFORM A 300 MICRO SECOND
1$:     DCR     B               ;TIMEOUT
        JNZ     1$
        IN      RSTAT           ;GET THE READ PATH STATUS
        CPI     GOODTM          ;GOOD STATUS?
        JZ      AMDCNJ          ;YES-CONTINUE
        ROUT    ADATA           ;NO-ERROR-STORE ACTUAL STATUS
                OUT     ADATA           ;WRITE AC INTO ADATA
                MOV     A,A             ;RETRY LINK
        MVI     A,GOODTM        ;GET EXPECTED STATUS
        ROUT    EDATA           ;STORE EXPECTED STATUS
                OUT     EDATA           ;WRITE AC INTO EDATA
                MOV     A,A             ;RETRY LINK
```

```
2175  478C                                                ERRB    AMDLPJ,AMDCNJ,
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  478C  CD  12  28        18.0                                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000D                                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  478F  0D                                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4790  00                                                     .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4791  CD  15  28        18.0                         AMDCNJ::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4794  DA  6F  47        10.0                                 JC      AMDLPJ          ;LOOP ADDRESS IF LOOP SPECIFIED
2176                                                    ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD TAPE MARK
2177  4797  2A  FF  4A        16.0  2$:     LHLD    AMTMSK          ;INCREMENT THE AMTIE
2178  479A  23                  6.0          INX     H               ;MASK BY 1
2179  479B  22  FF  4A        16.0          SHLD    AMTMSK          ;SAVE THE MASK
2180  479E  3A  00  4B        13.0          LDA     AMTMSK+1
2181  47A1  FE  01             7.0          CPI     @1
2182  47A3  CA  1D  47        10.0          JZ      AMDLPI
2183  47A6  3A  FF  4A        13.0          LDA     AMTMSK          ;GET LOW ORDER 8 BITS OF AMTIE MASK
2184  47A9  FE  58             7.0          CPI     @130            ;EQUAL A GOOD TAPE MARK
2185  47AB  CA  97  47        10.0          JZ      2$              ;YES-INCREMENT AGAIN
2186  47AE  3A  00  4B        13.0          LDA     AMTMSK+1        ;NO-END OF TEST
2187  47B1  FE  02             7.0          CPI     @2              ;
2188  47B3  C2  1D  47        10.0          JNZ     AMDLPI          ;NO-KEEP ON GOING
2189  47B6                                   ENDTST  TST06X          ;YES-
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47B6                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47B6  CD  06  28        18.0                         CALL    REQST
 (2)  47B9  00                                             .BYTE                   ;DATA PATTERN NUMBER
 (2)  47BA  00  00                                         .WORD                   ;SYSTEM '*' COUNT
 (2)  47BC  00  00                                         .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47BE  00                                             .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  47BF  07                                             .BYTE   7               ;REQUEST CODE
 (1)  47C0  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  47C3  3D                  4.0          DCR     A               ;DOWNCOUNT
 (1)  47C4  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  47C7  F2  0E  47        10.0          JP      TST06X          ;DO TEST UNTIL TILL = 0
2190  47CA  CD  CD  48        18.0          CALL    CLEAR
```

```
                                .SBTTL  TEST 07 - 5 X 4 TRANSLATION TEST (FORWARD)
                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*5 X 4 TRANSLATION TEST (FORWARD)
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
                                ;*AS WELL AS THE READ CHANNELS ABILITY TO 5 TO 4 TRANSLATE ALL POSSIBLE
                                ;*COMBINATIONS OF INPUT DATA IN THE FORWARD MODE.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*     DELAY 50 MICROSECONDS
                                ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*     CLEAR THE TIE BUS
                                ;*     SET UP INPUT TABLE POINTER
                                ;*     SET UP OUTPUT TABLE POINTER
                                ;*   BGNDO
                                ;*   : ENABLE THE READ PATH CLOCK
                                ;*   : SET PLO BYPASS MODE 2,TIE BUS JAM AND DISABLE PLO
                                ;*   : CLOCK THE FIFO
                                ;*   : ISSUE CLEAR ALL COMMAND
                                ;*   : WAIT
                                ;*   : STOP THE READ PATH
                                ;*   : PUT THE CONTROLLER IN GCR MODE
                                ;*   : INIT THE SUB-GROUP COUNT TO 2
                                ;*   : BGNDO
                                ;*   : : INIT THE LOOP COUNT TO 5
                                ;*   : : BGNDO
                                ;*   : : :     MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
                                ;*   : : :     INCREMENT THE INPUT TABLE POINTER
                                ;*   : : :     MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
                                ;*   : : :     CLOCK THE DATA INTO THE FIFO
                                ;*   : : :     DECREMENT THE LOOP COUNT
                                ;*   : : :     INCREMENT THE INPUT TABLE POINTER
                                ;*   : : :     DO UNTIL LOOP COUNT = 0
                                ;*   : : ENDDO
                                ;*   : : DECREMENT THE SUB-GROUP COUNTER
                                ;*   : : DO UNTIL THE SUB-GROUP COUNTER =0
                                ;*   : ENDDO
                                ;*   . ISSUE DIAGNOSTIC READ COMMAND
                                ;*   : SET UP WATCHDOG TIMER COUNT
                                ;*   : BGNDO
                                ;*   : :     SINGLE STEP THE READ PATH
```

```
                                ;*  : :    DECREMENT THE WATCHDOG TIMER
                                ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
                                ;*  : ENDDO
                                ;*  : IF WATCHDOG TIMER=0
                                ;*  : :    THEN-ERROR
                                ;*  : :    ELSE-CONTINUE
                                ;*  : ENDIF
                                ;*  . SINGLE STEP THE READ PATH
                                ;*  : INIT THE LOOP COUNT TO 4
                                ;*  : BGNDO
                                ;*  : :    SINGLE STEP THE READ PATH
                                ;*  : :    INPUT THE ACTUAL DATA
                                ;*  : :    INPUT THE ACTUAL PARITY
                                ;*  : :    GET THE EXPECTED DATA FROM THE EXPECTED DATA TABLE
                                ;*  : :    INCREMENT THE OUTPUT TABLE POINTER
                                ;*  : :    GET THE EXPECTED PARITY FROM THE EXPECTED DATA TABLE
                                ;*  : :    INCREMENT THE OUTPUT TABLE POINTER
                                ;*  : :    CALL SUBROUTINE ''CK5X4''
                                ;*  : :    DECREMENT THE LOOP COUNT
                                ;*  : :    DO UNTIL LOOP COUNT = 0
                                ;*  : ENDDO
                                ;*  :  DO UNTIL END OF INPUT DATA TABLE
                                ;*  ENDDO
                                ;*ENDTST
2261  47CD                      SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
2262                            ;*RPM3 MICRO TEST 07
2263                            ;*RPM3 MICRO ERROR 16
2264                            ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2265                            ;*M8950'S, M8953
2266                            ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
2267                            ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2268                            ;*FATAL ERROR - MICRO TEST ABORTED
2269                            ;*
2270                            ;*RPM3 MICRO TEST 07
2271                            ;*RPM3 MICRO ERROR 17
2272                            ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2273                            ;*M8950'S, M8953
2274                            ;*CHANNEL 0 FAILED
2275                            ;*ACTUAL = NNNN
2276                            ;*EXPECTED = NNNN
2277                            ;*SUBGROUP NUMBER = LLL
2278                            ;*
2279                            ;*RPM3 MICRO TEST 07
2280                            ;*RPM3 MICRO ERROR 20
2281                            ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2282                            ;*M8950'S, M8953
2283                            ;*CHANNEL 1 FAILED
2284                            ;*ACTUAL = NNNN
2285                            ;*EXPECTED - NNNN
2286                            ;*SUBGROUP NUMBER - LLL
2287                            ;*
```

```
2288                                    ;*RPM3 MICRO TEST 07
2289                                    ;*RPM3 MICRO ERROR 21
2290                                    ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2291                                    ;*M8950'S, M8953
2292                                    ;*CHANNEL 2 FAILED
2293                                    ;*ACTUAL = NNNN
2294                                    ;*EXPECTED = NNNN
2295                                    ;*SUBGROUP NUMBER = LLL
2296                                    ;*
2297                                    ;*RPM3 MICRO TEST 07
2298                                    ;*RPM3 MICRO ERROR 22
2299                                    ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2300                                    ;*M8950'S, M8953
2301                                    ;*CHANNEL 3 FAILED
2302                                    ;*ACTUAL = NNNN
2303                                    ;*EXPECTED = NNNN
2304                                    ;*SUBGROUP NUMBER = LLL
2305                                    ;*
2306                                    ;*RPM3 MICRO TEST 07
2307                                    ;*RPM3 MICRO ERROR 23
2308                                    ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2309                                    ;*M8950'S, M8953
2310                                    ;*CHANNEL 4 FAILED
2311                                    ;*ACTUAL = NNNN
2312                                    ;*EXPECTED = NNNN
2313                                    ;*SUBGROUP NUMBER = LLL
2314                                    ;*
2315                                    ;*RPM3 MICRO TEST 07
2316                                    ;*RPM3 MICRO ERROR 24
2317                                    ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2318                                    ;*M8950'S, M8953
2319                                    ;*CHANNEL 5 FAILED
2320                                    ;*ACTUAL = NNNN
2321                                    ;*EXPECTED = NNNN
2322                                    ;*SUBGROUP NUMBER = LLL
2323                                    ;*
2324                                    ;*RPM3 MICRO TEST 07
2325                                    ;*RPM3 MICRO ERROR 25
2326                                    ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2327                                    ;*M8950'S, M8953
2328                                    ;*CHANNEL 6 FAILED
2329                                    ;*ACTUAL = NNNN
2330                                    ;*EXPECTED = NNNN
2331                                    ;*SUBGROUP NUMBER = LLL
2332                                    ;*
2333                                    ;*RPM3 MICRO TEST 07
2334                                    ;*RPM3 MICRO ERROR 26
2335                                    ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2336                                    ;*M8950'S, M8953
2337                                    ;*CHANNEL 7 FAILED
2338                                    ;*ACTUAL = NNNN
2339                                    ;*EXPECTED = NNNN
2340                                    ;*SUBGROUP NUMBER - LLL
2341                                    ;*
```

```
                                        ;*RPM3 MICRO TEST 07
                                        ;*RPM3 MICRO ERROR 27
                                        ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*SUBGROUP NUMBER = LLL
47CD                                    S
 (1)                                    ; ************************************************************************************
                                        TEST7:  TESTX   @7              ;INITIALIZE THE TEST
 (1) 47CD 3E 07           7.0                           MVI     A,@7            ;DEFINE THE TEST NUMBER
 (1) 47CF CD 03 28       18.0                           CALL    TSET            ;SETUP THE TEST
                                        ;%RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                        ;&M8950'S, M8953

47D2 3E 10               7.0    TST07X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
47D4 D3 F0              10.0            OUT     CLKCTL
47D6 CD CD 48           18.0            CALL    CLEAR           ;CLEAR ALL TU PORTS
47D9 3A FE 4A           13.0            LDA     SUNIT
47DC D3 E0              10.0            OUT     MBSEL
47DE 3E 04               7.0            MVI     A,P.RPEN
47E0 D3 4C              10.0            OUT     PENAB
47E2 3E 60               7.0            MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
47E4 D3 48              10.0            OUT     PDIAG           ;LCS, LWR AND DPEN
47E6 AF                  4.0            XRA     A
47E7 D3 44              10.0            OUT     TAMT
47E9 3E CA               7.0            MVI     A,10            ;DELAY
47EB 3D                  4.0    1$:     DCR     A               ;50
47EC C2 EB 47           10.0            JNZ     1$              ;MICROSECONDS
47EF D3 D2              10.0            OUT     TRKENA          ;CLEAR ALL TRACKS FROM
47F1 D3 D2              10.0            OUT     TRKENA          ;THE TRANSLATOR

47F3 AF                  4.0            XRA     A               ;CLEAR THE TIE BUS
47F4 D3 0A              10.0            OUT     RTIEB
47F6 3E 01               7.0            MVI     A,@1            ;INITIALIZE THE SUBGROUP COUNT
47F8 32 0B 4B           13.0            STA     SUBCNT          ;

47FB 21 DD 49           10.0            LXI     H,M8950I        ;SET UP POINTER TO INPUT DATA TABLE
47FE 22 03 4B           16.0            SHLD    INTBL           ;SAVE IT
4801 21 7D 4A           10.0            LXI     H,M89500        ;SET UP POINTER TO OUTPUT DATA TABLE
4804 22 05 4B           16.0            SHLD    OUTTBL          ;SAVE IT
4807 3E A8               7.0    TST07C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
                                                                ;DISABLE THE PLO AND SET JAM TIE BUS
4809 D3 09              10.0            OUT     RPCTL
480B 3E 10               7.0            MVI     A,W.GCR         ;SET GCR MODE
480D D3 D3              10.0            OUT     WMCCTL
480F D3 08              10.0            OUT     RFIFOL          ;CLOCK THE FIFO
4811 3E 0D               7.0            MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
4813 D3 0B              10.0            OUT     RCMD
4815 00                  4.0            NOP                     ;WAIT
4816 00                  4.0            NOP
4817 00                  4.0            NOP
```

```
2393  4818  00              4.0          NOP
2394  4819  3E  A9          7.0          MVI   A,R.PLO1!R.STPC.R.PLOD!R.TBJN   ;STOP THE READ PATH
2395  481B  D3  09         10.0          OUT   RPCTL
2396
2397  481D  0E  02          7.0          MVI   C,2              ;INITIALIZE THE SUB-GROUP COUNTER
2398  481F  2A  03  4B      16.0  3$:     LHLD  INTBL            ;GET POINTER TO INPUT DATA TABLE
2399  4822  06  05          7.0          MVI   B,@5             ;SET UP LOOP COUNT
2400  4824  7E              7.0  1$:     MOV   A,M              ;GET A DATA BYTE
2401  4825  D3  40         10.0          OUT   TCMD             ;STORE DATA IN COMMAND ADDRESS
2402  4827  23              6.0          INX   H                ;POINT TO DATA PARITY
2403  4828  7E              7.0          MOV   A,M              ;GET THE DATA PARITY
2404  4829  07              4.0          RLC                    ;POSITION FOR OUTPUT
2405  482A  F6  60          7.0          ORI   P.LWR!P.LCS      ;OR IN CONTROL BITS
2406  482C  D3  48         10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
2407  482E  D3  08         10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFO'S
2408
2409  4830  23              6.0          INX   H                ;UPDATE THE TABLE POINTER
2410  4831  05              4.0          DCR   B                ;DECREMENT LOOP COUNT
2411  4832  C2  24  48      10.0          JNZ   1$               ;DO UNTIL LOOP COUNT = 0
2412  4835  0D              4.0          DCR   C                ;DECREMENT THE SUBGROUP COUNTER
2413  4836  C2  1F  48      10.0          JNZ   3$               ;LOAD THE SUB-GROUP TWICE
2414                                ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
2415
2416  4839  22  01  4B      16.0          SHLD  INTBLTMP         ;SAVE THE INPUT DATA TABLE POINTER
2417  483C  3E  0B          7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
2418  483E  D3  0B         10.0          OUT   RCMD             ;COMMAND
2419  4840  11  01  00      10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
2420  4843  21  A8  FD      10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
2421  4846  D3  0C         10.0  2$:     OUT   RINST            ;STEP THE READ PATH
2422  4848  DB  01         10.0          IN    RPCHI            ;DATA READY SET?
2423  484A  E6  10          7.0          ANI   R.DRDY
2424  484C  C2  61  48      10.0          JNZ   TST07A           ;YES-GO PROCESS
2425  484F  19              10.0          DAD   D                ;WATCH DOG TIMEOUT?
2426  4850  D2  46  48      10.0          JNC   2$               ;NO-CONTINUE
2427  4853                               ERR   EXIT,DUMMX
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4853  CD  09  28        18.0                    CALL  ERLP           ;PROCESS ERROR - DO 2.3
 (1)       000E                                     MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4856  0E                                      .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  4857  00                                      .BYTE
 (1)  4858  CD  15  28        18.0            DUMMX:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  485B  DA  CA  48        10.0                    JC    EXIT           ;LOOP ADDRESS IF LOOP SPECIFIED
2428                          ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2429                          ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2430                          ;<FATAL ERROR - MICRO TEST ABORTED
2431  485E  C3  CA  48        10.0          JMP   EXIT
2432
2433  4861  2A  05  4B      16.0  TST07A: LHLD  OUTTBL          ;GET ADDRESS OF EXPECTED OUTPUT DATA TABLE
2434  4864  00              4.0          NOP
2435  4865  00              4.0          NOP
2436  4866  00              4.0          NOP
2437  4867  0E  04          7.0          MVI   C,@4             ;INIT THE LOOP COUNT
2438  4869  00              4.0  TST07B: NOP
2439  486A  D3  0C         10.0          OUT   RINST            ;SINGLE STEP THE READ PATH
```

```
2440  486C  DB  17      10.0          IN      RDATA           ;INPUT THE ACTUAL DATA BYTE
2441  486E  32  07  4B  13.0          STA     DATAA           ;STORE IT
2442  4871  DB  15      10.0          IN      RPSTA           ;INPUT THE ACTUAL PARITY BIT
2443  4873  E6  40       7.0          ANI     R.DATA          ;REMOVE JUNK BITS
2444  4875  07           4.0          RLC                     ;POSITION THE BIT FOR COMPARE
2445  4876  07           4.0          RLC
2446  4877  32  08  4B  13.0          STA     DATAAP          ;STORE IT
2447  487A  7E           7.0          MOV     A,M             ;GET THE EXPECTED DATA BYTE
2448  487B  32  09  4B  13.0          STA     DATAE           ;STORE IT
2449  487E  23           6.0          INX     H               ;UPDATE THE POINTER
2450  487F  7E           7.0          MOV     A,M             ;GET THE EXPECTED PARITY BYTE
2451  4880  32  0A  4B  13.0          STA     DATAEP          ;STORE IT
2452  4883  EB           4.0          XCHG                    ;SAVE H AND L IN D AND E
2453  4884  3A  0B  4B  13.0          LDA     SUBCNT          ;GET THE SUBGROUP COUNT
2454  4887                           ROUT    R05L            ;STORE IN CAS
 (1)  4887  D3  8A      10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  4889  7F           4.0                  MOV     A,A             ;RETRY LINK
2455  488A  AF           4.0          XRA     A               ;CLEAR THE HIGH HALF OF THE SUBGROUP COUNTER
2456  488B                           ROUT    R05H            ;
 (1)  488B  D3  8B      10.0                  OUT     R05H            ;WRITE AC INTO R05H
 (1)  488D  7F           4.0                  MOV     A,A             ;RETRY LINK
2457  488E  CD  F8  48  18.0          CALL    CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
2458  4891  DA  07  48  10.0          JC      TST07C          ;IF LOOP ON ERROR REPEAT THE TEST
2459  4894  EB           4.0          XCHG                    ;RESTORE H AND L FROM D AND E
2460  4895  23           6.0          INX     H               ;UPDATE THE EXPECTED DATA TABLE POINTER
2461  4896  0D           4.0          DCR     C               ;DECREMENT THE LOOP COUNT
2462  4897  C2  69  48  10.0          JNZ     TST07B          ;DO UNTIL LOOP COUNT = ZERO
2463  489A  22  05  4B  16.0          SHLD    OUTTBL          ;SAVE EXPECTED DATA TABLE POINTER
2464  489D  3A  0B  4B  13.0          LDA     SUBCNT          ;GET THE SUBGROUP COUNTER
2465  48A0  3C           4.0          INR     A               ;UPDATE FOR THE NEXT SUBGROUP
2466  48A1  32  08  4B  13.0          STA     SUBCNT          ;SAVE IT
2467  48A4  2A  01  4B  16.0          LHLD    INTBLTMP        ;GET THE INPUT TABLE POINTER
2468  48A7  22  03  4B  16.0          SHLD    INTBL           ;UPDATE THE INPUT TABLE POINTER
2469  48AA  2B           6.0          DCX     H               ;UPDATE TO LAST PARITY BYTE
2470  48AB  7E           7.0          MOV     A,M             ;GET LAST PARITY BYTE
2471  48AC  A7           4.0          ANA     A               ;SET CONDITION BITS
2472  48AD  F2  07  48  10.0          JP      TST07C          ;CONTINUE UNTIL END OF INPUT TABLE
2473  48B0  CD  CD  48  18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
2474  48B3                           ENDTST  TST07X
 (1)                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  48B3                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  48B3  CD  06  28  18.0                          CALL    REQST
 (2)  48B6  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  48B7  00  00                                    .WORD                   ;SYSTEM '*' COUNT
 (2)  48B9  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  48BB  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  48BC  07                                        .BYTE   7               ;REQUEST CODE
 (1)  48BD  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  48C0  3D           4.0          DCR     A               ;DOWNCOUNT
 (1)  48C1  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  48C4  F2  D2  47  10.0          JP      TST07X          ;DO TEST UNTIL TILL - 0
2475  48C7  CD  CD  48  18.0          CALL    CLEAR
2476
2477  48CA  C3  18  28  10.0  EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************
48CD F5                      12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
48CE C5                      12.0          PUSH    B
48CF 06 00                    7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
48D1 DB E0                   10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
48D3 E6 80                    7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
48D5 B0                       4.0          ORA     B               ;ADD IN THE SELECTED PORT #
48D6 D3 E0                   10.0          OUT     MBSEL           ;RESET TO THIS PORT
48D8 3E 80                    7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
48DA D3 40                   10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
48DC AF                       4.0          XRA     A               ;CLEAR TU COMMAND A
48DD D3 40                   10.0          OUT     TCMD            ;
48DF 3E 81                    7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
48E1 D3 40                   10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
48E3 3E 00                    7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
48E5 D3 40                   10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
48E7 AF                       4.0          XRA     A
48E8 D3 44                   10.0          OUT     TAMT            ;CLEAR AMTIE WORD
48EA D3 48                   10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
48EC D3 4C                   10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
48EE 04                       4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
48EF 78                       4.0          MOV     A,B
48F0 FE 04                    7.0          CPI     4               ;DONE?
48F2 C2 D1 48                10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
48F5 C1                      10.0          POP     B               ;RESET B & C
48F6 F1                      10.0          POP     PSW             ;ALL DONE
48F7 C9                      10.0          RET                     ;EXIT
```

```
2523                                .SBTTL  SUBROUTINE CHECK 5X4
2524  48F8                          SSUB
 (1)                                ;*****************************************************************************
 (1)                                ;*SUBROUTINE TITLE
 (1)                                ;*---------- -----
2525                                ;*CHECK 5X4 SUBROUTINE
2526  48F8                          SD
 (1)                                ;*****************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
2527                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ CHANNEL DATA
2528                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
2529                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
2530  48F8                          SP
 (1)                                ;*****************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
2531                                ;*BGNSUB
2532                                ;*     IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
2533                                ;*       THEN-ERROR CHANNEL 0
2534                                ;*       ELSE-CONTINUE
2535                                ;*     ENDIF
2536                                ;*     IF CHANNEL 1 ACTUAL INFORMATION NOT - CHANNEL 1 EXPECTED
2537                                ;*       THEN-ERROR CHANNEL 1
2538                                ;*       ELSE-CONTINUE
2539                                ;*     ENDIF
2540                                ;*     IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
2541                                ;*       THEN-ERROR CHANNEL 2
2542                                ;*       ELSE-CONTINUE
2543                                ;*     ENDIF
2544                                ;*     IF CHANNEL 3 ACTUAL INFORMATION NOT - CHANNEL 3 EXPECTED
2545                                ;*       THEN-ERROR CHANNEL 3
2546                                ;*       ELSE-CONTINUE
2547                                ;*     ENDIF
2548                                ;*     IF CHANNEL 4 ACTUAL INFORMATION NOT - CHANNEL 4 EXPECTED
2549                                ;*       THEN-ERROR CHANNEL 4
2550                                ;*       ELSE-CONTINUE
2551                                ;*     ENDIF
2552                                ;*     IF CHANNEL 5 ACTUAL INFORMATION NOT - CHANNEL 5 EXPECTED
2553                                ;*       THEN-ERROR CHANNEL 5
2554                                ;*       ELSE-CONTINUE
2555                                ;*     ENDIF
2556                                ;*     IF CHANNEL 6 ACTUAL INFORMATION NOT - CHANNEL 6 EXPECTED
2557                                ;*       THEN-ERROR CHANNEL 6
2558                                ;*       ELSE-CONTINUE
2559                                ;*     ENDIF
2560                                ;*     IF CHANNEL 7 ACTUAL INFORMATION NOT - CHANNEL 7 EXPECTED
2561                                ;*       THEN-ERROR CHANNEL 7
2562                                ;*       ELSE-CONTINUE
2563                                ;*     ENDIF
2564                                ;*     IF CHANNEL P ACTUAL INFORMATION NOT - CHANNEL P EXPECTED
2565                                ;*       THEN-ERROR CHANNEL P
2566                                ;*       ELSE-CONTINUE
2567                                ;*     ENDIF
```

```
                                        ;*ENDSUB
                                        $
                                        ; ***************************************************************************
                                        CK5X4:  LXI     H,DATAE         ;LOAD EXPECTED DATA ADDRESS
                                                LDA     DATAA           ;GET THE ACTUAL DATA
                                                ROUT    ADATA           ;WRITE THE ACTUAL DATA TO THE CAS
                                                        OUT     ADATA           ;WRITE AC INTO ADATA
                                                        MOV     A,A             ;RETRY LINK
                                                ANI     $01             ;REMOVE UNWANTED BITS
                                                MOV     B,A             ;SAVE IN B
                                                MOV     A,M             ;GET EXPECTED DATA
                                                ROUT    EDATA           ;WRITE THE EXPECTED DATA TO THE CAS
                                                        OUT     EDATA           ;WRITE AC INTO EDATA
                                                        MOV     A,A             ;RETRY LINK
                                                ANI     $01             ;REMOVE UNWANTED BITS
                                                CMP     B               ;COMPARE
                                                JZ      CKAM0           ;CONTINUE IF EQUAL
                                                ERRB    OUT,CKAM0,@7
                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                                .BYTE   @7              ;PRINT ROUTINE NUMBER
                                                        CKAM0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                                JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>CHANNEL 0 FAILED

                                                LDA     DATAA           ;GET THE ACTUAL DATA
                                                ANI     $02             ;REMOVE UNWANTED BITS
                                                MOV     B,A             ;SAVE IN B
                                                MOV     A,M             ;GET EXPECTED DATA
                                                ANI     $02             ;REMOVE UNWANTED BITS
                                                CMP     B               ;COMPARE
                                                JZ      CKAM1           ;CONTINUE IF EQUAL
                                                ERRB    OUT,CKAM1,@7
                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
                                                                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                                .BYTE   @7              ;PRINT ROUTINE NUMBER
                                                        CKAM1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                                                JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>CHANNEL 1 FAILED
```

```
2593  4931  3A 07 4B      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2594  4934  E6 04           7.0          ANI     $04             ;REMOVE UNWANTED BITS
2595  4936  47              4.0          MOV     B,A             ;SAVE IN B
2596  4937  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2597  4938  E6 04           7.0          ANI     $04             ;REMOVE UNWANTED BITS
2598  493A  B8              4.0          CMP     B               ;COMPARE
2599  493B  CA 43 49       10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
2600  493E                               ERRB    OUT,CKAM2,@7
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  493E  CD 12 28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0011                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4941    11                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4942    07                              .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4943  CD 15 28       18.0          CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4946  DA DC 49       10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2601                                ;>CHANNEL 2 FAILED
2602
2603  4949  3A 07 4B      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2604  494C  E6 08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
2605  494E  47              4.0          MOV     B,A             ;SAVE IN B
2606  494F  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2607  4950  E6 08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
2608  4952  B8              4.0          CMP     B               ;COMPARE
2609  4953  CA 5B 49       10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2610  4956                               ERRB    OUT,CKAM3,@7
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4956  CD 12 28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0012                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4959    12                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  495A    07                              .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  495B  CD 15 28       18.0          CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  495E  DA DC 49       10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2611                                ;>CHANNEL 3 FAILED
2612
2613  4961  3A 07 4B      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2614  4964  E6 10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
2615  4966  47              4.0          MOV     B,A             ;SAVE IN B
2616  4967  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2617  4968  E6 10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
2618  496A  B8              4.0          CMP     B               ;COMPARE
2619  496B  CA 73 49       10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2620  496E                               ERRB    OUT,CKAM4,@7
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  496E  CD 12 28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0013                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4971    13                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4972    07                              .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4973  CD 15 28       18.0          CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4976  DA DC 49       10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2621                                ;>CHANNEL 4 FAILED
```

```
2623  4979  3A  07  4B        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2624  497C  E6  20              7.0          ANI     $20             ;REMOVE UNWANTED BITS
2625  497E  47                  4.0          MOV     B,A             ;SAVE IN B
2626  497F  7E                  7.0          MOV     A,M             ;GET EXPECTED DATA
2627  4980  E6  20              7.0          ANI     $20             ;REMOVE UNWANTED BITS
2628  4982  B8                  4.0          CMP     B               ;COMPARE
2629  4983  CA  88  49         10.0          JZ      CKAM5           ;CONTINUE IF FQUAL
2630  4986                                   ERRB    OUT,CKAM5,@7
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4986  CD  12  28         18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0014                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4989  14                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  498A  07                                       .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  498B  CD  15  28         18.0           CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  498E  DA  DC  49         10.0                   JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2631                                     ;>CHANNEL 5 FAILED
2632
2633  4991  3A  07  4B        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2634  4994  E6  40              7.0          ANI     $40             ;REMOVE UNWANTED BITS
2635  4996  47                  4.0          MOV     B,A             ;SAVE IN B
2636  4997  7E                  7.0          MOV     A,M             ;GET EXPECTED DATA
2637  4998  E6  40              7.0          ANI     $40             ;REMOVE UNWANTED BITS
2638  499A  B8                  4.0          CMP     B               ;COMPARE
2639  499B  CA  A3  49         10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
2640  499E                                   ERRB    OUT,CKAM6,@7
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  499E  CD  12  28         18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0015                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49A1  15                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49A2  07                                       .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  49A3  CD  15  28         18.0           CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49A6  DA  DC  49         10.0                   JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2641                                     ;>CHANNEL 6 FAILED
2642
2643  49A9  3A  07  4B        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2644  49AC  E6  80              7.0          ANI     $80             ;REMOVE UNWANTED BITS
2645  49AE  47                  4.0          MOV     B,A             ;SAVE IN B
2646  49AF  7E                  7.0          MOV     A,M             ;GET EXPECTED DATA
2647  49B0  E6  80              7.0          ANI     $80             ;REMOVE UNWANTED BITS
2648  49B2  B8                  4.0          CMP     B               ;COMPARE
2649  49B3  CA  BB  49         10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
2650  49B6                                   ERRB    OUT,CKAM7,@7
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  49B6  CD  12  28         18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0016                                      MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49B9  16                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49BA  07                                       .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  49BB  CD  15  28         18.0           CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49BE  DA  DC  49         10.0                   JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2651                                     ;>CHANNEL 7 FAILED
```

```
2653  49C1  21  0A  4B      10.0          LXI     H,DATAEP          ;GET THE EXPECTED DATA PARITY ADDRESS
2654  49C4  3A  08  4B      13.0          LDA     DATAAP            ;GET THE ACTUAL DATA
2655  49C7  E6  01           7.0          ANI     $01               ;REMOVE UNWANTED BITS
2656  49C9  47               4.0          MOV     B,A               ;SAVE IN B
2657  49CA  7E               7.0          MOV     A,M               ;GET EXPECTED DATA
2658  49CB  E6  01           7.0          ANI     $01               ;REMOVE UNWANTED BITS
2659  49CD  B8               4.0          CMP     B                 ;COMPARE
2660  49CE  CA  D6  49       10.0         JZ      CKAMP             ;CONTINUE IF EQUAL
2661  49D1                                ERRB    OUT,CKAMP,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  49D1  CD  12  28      18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)       0017                                   MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49D4  17                                    .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  49D5  07                                    .BYTE   @7                ;PRINT ROUTINE NUMBER
 (1)  49D6  CD  15  28      18.0          CKAMP:: CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49D9  DA  DC  49      10.0                  JC      OUT               ;LOOP ADDRESS IF LOOP SPECIFIED
2662                                ;>CHANNEL P FAILED
2663
2664  49DC  C9               10.0  OUT:    RET
```

```
                                .SBTTL  TABLE UNTRANSLATED INPUT DATA

                                ;INPUT DATA TABLE - THIS DATA IS LOADED INTO THE READ CHANNEL FIFO'S
                                ;ONE GROUP AT A TIME AND THEN TRANSLATED.

49DD  FF        M8950I: .BYTE   @377            ;DATA GROUP 1 - CHARACTER 1
49DE  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 1  PARITY
49DF  FF                .BYTE   @377            ;DATA GROUP 1 - CHARACTER 2
49E0  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 2  PARITY
49E1  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3
49E2  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3  PARITY
49E3  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4
49E4  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4  PARITY
49E5  FF                .BYTE   @377            ;DATA GROUP 1 - CHARACTER 5
49E6  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 5  PARITY

49E7  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 1
49E8  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 1  PARITY
49E9  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 2
49EA  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 2  PARITY
49EB  00                .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3
49EC  00                .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3  PARITY
49ED  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 4
49EE  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 4  PARITY
49EF  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 5
49F0  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 5  PARITY

49F1  FF                .BYTE   @377            ;DATA GROUP 3 - CHARACTER 1
49F2  01                .BYTE   @001            ;DATA GROUP 3 - CHARACTER 1  PARITY
49F3  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2
49F4  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2  PARITY
49F5  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3
49F6  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3  PARITY
49F7  FF                .BYTE   @377            ;DATA GROUP 3 - CHARACTER 4
49F8  01                .BYTE   @001            ;DATA GROUP 3 - CHARACTER 4  PARITY
49F9  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5
49FA  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5  PARITY

49FB  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 1
49FC  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 1  PARITY
49FD  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2
49FE  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2  PARITY
49FF  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3
4A00  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3  PARITY
4A01  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 4
4A02  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 4  PARITY
4A03  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 5
4A04  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 5  PARITY

4A05  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 1
4A06  01                .BYTE   @001            ;DATA GROUP 5 - CHARACTER 1  PARITY
4A07  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 2
4A08  01                .BYTE   @001            ;DATA GROUP 5 - CHARACTER 2  PARITY
4A09  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 3
```

```
2720  4A0A  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 3  PARITY
2721  4A0B  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4
2722  4A0C  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4  PARITY
2723  4A0D  FF          .BYTE   @377          ;DATA GROUP 5 - CHARACTER 5
2724  4A0E  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 5  PARITY
2725
2726  4A0F  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 1
2727  4A10  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 1  PARITY
2728  4A11  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2
2729  4A12  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2  PARITY
2730  4A13  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 3
2731  4A14  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 3  PARITY
2732  4A15  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4
2733  4A16  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4  PARITY
2734  4A17  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 5
2735  4A18  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 5  PARITY
2736
2737  4A19  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 1
2738  4A1A  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 1  PARITY
2739  4A1B  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2
2740  4A1C  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2  PARITY
2741  4A1D  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 3
2742  4A1E  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 3  PARITY
2743  4A1F  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 4
2744  4A20  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 4  PARITY
2745  4A21  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5
2746  4A22  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5  PARITY
2747
2748  4A23  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 1
2749  4A24  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 1  PARITY
2750  4A25  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2
2751  4A26  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2  PARITY
2752  4A27  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 3
2753  4A28  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 3  PARITY
2754  4A29  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 4
2755  4A2A  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 4  PARITY
2756  4A2B  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 5
2757  4A2C  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 5  PARITY
2758
2759  4A2D  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 1
2760  4A2E  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 1  PARITY
2761  4A2F  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 2
2762  4A30  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 2  PARITY
2763  4A31  00          .BYTE   @C00          ;DATA GROUP 9 - CHARACTER 3
2764  4A32  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3  PARITY
2765  4A33  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 4
2766  4A34  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 4  PARITY
2767  4A35  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5
2768  4A36  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5  PARITY
2769
2770  4A37  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1
2771  4A38  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1  PARITY
2772  4A39  FF          .BYTE   @377          ;DATA GROUP 10 - CHARACTER 2
2773  4A3A  01          .BYTE   @001          ;DATA GROUP 10 - CHARACTER 2  PARITY
```

```
2774  4A3B  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 3
2775  4A3C  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 3  PARITY
2776  4A3D  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 4
2777  4A3E  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 4  PARITY
2778  4A3F  FF          .BYTE   @377           ;DATA GROUP 10 - CHARACTER 5
2779  4A40  01          .BYTE   @001           ;DATA GROUP 10 - CHARACTER 5  PARITY
2780
2781  4A41  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 1
2782  4A42  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 1  PARITY
2783  4A43  FF          .BYTE   @377           ;DATA GROUP 11 - CHARACTER 2
2784  4A44  01          .BYTE   @001           ;DATA GROUP 11 - CHARACTER 2  PARITY
2785  4A45  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3
2786  4A46  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3  PARITY
2787  4A47  FF          .BYTE   @377           ;DATA GROUP 11 - CHARACTER 4
2788  4A48  01          .BYTE   @001           ;DATA GROUP 11 - CHARACTER 4  PARITY
2789  4A49  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5
2790  4A4A  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5  PARITY
2791
2792  4A4B  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 1
2793  4A4C  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 1  PARITY
2794  4A4D  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 2
2795  4A4E  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 2  PARITY
2796  4A4F  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3
2797  4A50  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3  PARITY
2798  4A51  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 4
2799  4A52  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 4  PARITY
2800  4A53  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 5
2801  4A54  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 5  PARITY
2802
2803  4A55  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 1
2804  4A56  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 1  PARITY
2805  4A57  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 2
2806  4A58  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 2  PARITY
2807  4A59  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 3
2808  4A5A  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 3  PARITY
2809  4A5B  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 4
2810  4A5C  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 4  PARITY
2811  4A5D  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5
2812  4A5E  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5  PARITY
2813
2814  4A5F  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 1
2815  4A60  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 1  PARITY
2816  4A61  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 2
2817  4A62  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 2  PARITY
2818  4A63  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 3
2819  4A64  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 3  PARITY
2820  4A65  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4
2821  4A66  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4  PARITY
2822  4A67  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 5
2823  4A68  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 5  PARITY
2824
2825  4A69  00          .BYTE   @000           ;DATA GROUP 15 - CHARACTER 1
2826  4A6A  00          .BYTE   @000           ;DATA GROUP 15 - CHARACTER 1  PARITY
2827  4A6B  FF          .BYTE   @377           ;DATA GROUP 15 - CHARACTER 2
```

```
2828  4A6C  01          .BYTE   @001           ;DATA GROUP 15 - CHARACTER 2  PARITY
2829  4A6D  FF          .BYTE   @377           ;DATA GROUP 15 - CHARACTER 3
2830  4A6E  01          .BYTE   @001           ;DATA GROUP 15 - CHARACTER 3  PARITY
2831  4A6F  FF          .BYTE   @377           ;DATA GROUP 15 - CHARACTER 4
2832  4A70  01          .BYTE   @001           ;DATA GROUP 15 - CHARACTER 4  PARITY
2833  4A71  00          .BYTE   @000           ;DATA GROUP 15 - CHARACTER 5
2834  4A72  00          .BYTE   @000           ;DATA GROUP 15 - CHARACTER 5  PARITY
2835
2836  4A73  00          .BYTE   @000           ;DATA GROUP 16 - CHARACTER 1
2837  4A74  00          .BYTE   @000           ;DATA GROUP 16 - CHARACTER 1  PARITY
2838  4A75  FF          .BYTE   @377           ;DATA GROUP 16 - CHARACTER 2
2839  4A76  01          .BYTE   @001           ;DATA GROUP 16 - CHARACTER 2  PARITY
2840  4A77  FF          .BYTE   @377           ;DATA GROUP 16 - CHARACTER 3
2841  4A78  01          .BYTE   @001           ;DATA GROUP 16 - CHARACTER 3  PARITY
2842  4A79  FF          .BYTE   @377           ;DATA GROUP 16 - CHARACTER 4
2843  4A7A  01          .BYTE   @001           ;DATA GROUP 16 - CHARACTER 4  PARITY
2844  4A7B  FF          .BYTE   @377           ;DATA GROUP 16 - CHARACTER 5
2845  4A7C  81          .BYTE   @201           ;DATA GROUP 16 - CHARACTER 5  PARITY
2846
```

```
                                        .SBTTL  TABLE TRANSLATED OUTPUT DATA
                                        ;OUTPUT DATA TABLE - THIS DATA IS COMPARED WITH THE OUTPUT OF THE READ
                                        ;CHANNEL BOARDS WHEN DATA READY SETS.  IT REFLECTS THE TRANSLATED INPUT
                                        ;DATA ON SUBGROUP BASIS.

4A7D  00                        M89500: .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 1
4A7E  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 1 PARITY
4A7F  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 2
4A80  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 2 PARITY
4A81  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3
4A82  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3 PARITY
4A83  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4
4A84  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4 PARITY

4A85  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 1
4A86  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 1 PARITY
4A87  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2
4A88  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2 PARITY
4A89  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3
4A8A  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3 PARITY
4A8B  FF                                .BYTE   @377            ;DATA GROUP 2 - DATA VALUE 4
4A8C  01                                .BYTE   @001            ;DATA GROUP 2 - DATA VALUE 4 PARITY

4A8D  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 1
4A8E  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 1 PARITY
4A8F  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2
4A90  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2 PARITY
4A91  FF                                .BYTE   @377            ;DATA GROUP 3 - DATA VALUE 3
4A92  01                                .BYTE   @001            ;DATA GROUP 3 - DATA VALUE 3 PARITY
4A93  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 4
4A94  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 4 PARITY

4A95  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1
4A96  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1 PARITY
4A97  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2
4A98  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2 PARITY
4A99  FF                                .BYTE   @377            ;DATA GROUP 4 - DATA VALUE 3
4A9A  01                                .BYTE   @001            ;DATA GROUP 4 - DATA VALUE 3 PARITY
4A9B  FF                                .BYTE   @377            ;DATA GROUP 4 - DATA VALUE 4
4A9C  01                                .BYTE   @001            ;DATA GROUP 4 - DATA VALUE 4 PARITY

4A9D  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 1
4A9E  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 1 PARITY
4A9F  FF                                .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 2
4AA0  01                                .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 2 PARITY
4AA1  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 3
4AA2  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 3 PARITY
4AA3  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4
4AA4  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4 PARITY

4AA5  00                                .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 1
4AA6  00                                .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 1 PARITY
4AA7  FF                                .BYTE   @377            ;DATA GROUP 6 - DATA VALUE 2
4AA8  01                                .BYTE   @001            ;DATA GROUP 6 - DATA VALUE 2 PARITY
```

```
2902  4AA9  00          .BYTE   @000        ;DATA GROUP 6 - DATA VALUE 3
2903  4AAA  00          .BYTE   @000        ;DATA GROUP 6 - DATA VALUE 3 PARITY
2904  4AAB  FF          .BYTE   @377        ;DATA GROUP 6 - DATA VALUE 4
2905  4AAC  01          .BYTE   @001        ;DATA GROUP 6 - DATA VALUE 4 PARITY
2906
2907  4AAD  00          .BYTE   @000        ;DATA GROUP 7 - DATA VALUE 1
2908  4AAE  00          .BYTE   @000        ;DATA GROUP 7 - DATA VALUE 1 PARITY
2909  4AAF  FF          .BYTE   @377        ;DATA GROUP 7 - DATA VALUE 2
2910  4AB0  01          .BYTE   @001        ;DATA GROUP 7 - DATA VALUE 2 PARITY
2911  4AB1  FF          .BYTE   @377        ;DATA GROUP 7 - DATA VALUE 3
2912  4AB2  01          .BYTE   @001        ;DATA GROUP 7 - DATA VALUE 3 PARITY
2913  4AB3  00          .BYTE   @000        ;DATA GROUP 7 - DATA VALUE 4
2914  4AB4  00          .BYTE   @000        ;DATA GROUP 7 - DATA VALUE 4 PARITY
2915
2916  4AB5  00          .BYTE   @000        ;DATA GROUP 8 - DATA VALUE 1
2917  4AB6  00          .BYTE   @000        ;DATA GROUP 8 - DATA VALUE 1 PARITY
2918  4AB7  FF          .BYTE   @377        ;DATA GROUP 8 - DATA VALUE 2
2919  4AB8  01          .BYTE   @001        ;DATA GROUP 8 - DATA VALUE 2 PARITY
2920  4AB9  FF          .BYTE   @377        ;DATA GROUP 8 - DATA VALUE 3
2921  4ABA  01          .BYTE   @001        ;DATA GROUP 8 - DATA VALUE 3 PARITY
2922  4ABB  FF          .BYTE   @377        ;DATA GROUP 8 - DATA VALUE 4
2923  4ABC  01          .BYTE   @001        ;DATA GROUP 8 - DATA VALUE 4 PARITY
2924
2925  4ABD  FF          .BYTE   @377        ;DATA GROUP 9 - DATA VALUE 1
2926  4ABE  01          .BYTE   @001        ;DATA GROUP 9 - DATA VALUE 1 PARITY
2927  4ABF  00          .BYTE   @000        ;DATA GROUP 9 - DATA VALUE 2
2928  4AC0  00          .BYTE   @000        ;DATA GROUP 9 - DATA VALUE 2 PARITY
2929  4AC1  00          .BYTE   @000        ;DATA GROUP 9 - DATA VALUE 3
2930  4AC2  00          .BYTE   @000        ;DATA GROUP 9 - DATA VALUE 3 PARITY
2931  4AC3  00          .BYTE   @000        ;DATA GROUP 9 - DATA VALUE 4
2932  4AC4  00          .BYTE   @000        ;DATA GROUP 9 - DATA VALUE 4 PARITY
2933
2934  4AC5  FF          .BYTE   @377        ;DATA GROUP 10 - DATA VALUE 1
2935  4AC6  01          .BYTE   @001        ;DATA GROUP 10 - DATA VALUE 1 PARITY
2936  4AC7  00          .BYTE   @000        ;DATA GROUP 10 - DATA VALUE 2
2937  4AC8  00          .BYTE   @000        ;DATA GROUP 10 - DATA VALUE 2 PARITY
2938  4AC9  00          .BYTE   @000        ;DATA GROUP 10 - DATA VALUE 3
2939  4ACA  00          .BYTE   @000        ;DATA GROUP 10 - DATA VALUE 3 PARITY
2940  4ACB  FF          .BYTE   @377        ;DATA GROUP 10 - DATA VALUE 4
2941  4ACC  01          .BYTE   @001        ;DATA GROUP 10 - DATA VALUE 4 PARITY
2942
2943  4ACD  FF          .BYTE   @377        ;DATA GROUP 11 - DATA VALUE 1
2944  4ACE  01          .BYTE   @001        ;DATA GROUP 11 - DATA VALUE 1 PARITY
2945  4ACF  00          .BYTE   @000        ;DATA GROUP 11 - DATA VALUE 2
2946  4AD0  00          .BYTE   @000        ;DATA GROUP 11 - DATA VALUE 2 PARITY
2947  4AD1  FF          .BYTE   @377        ;DATA GROUP 11 - DATA VALUE 3
2948  4AD2  01          .BYTE   @001        ;DATA GROUP 11 - DATA VALUE 3 PARITY
2949  4AD3  00          .BYTE   @000        ;DATA GROUP 11 - DATA VALUE 4
2950  4AD4  00          .BYTE   @000        ;DATA GROUP 11 - DATA VALUE 4 PARITY
2951
2952  4AD5  FF          .BYTE   @377        ;DATA GROUP 12 - DATA VALUE 1
2953  4AD6  01          .BYTE   @001        ;DATA GROUP 12 - DATA VALUE 1 PARITY
2954  4AD7  00          .BYTE   @000        ;DATA GROUP 12 - DATA VALUE 2
2955  4AD8  00           BYTE   @000        ;DATA GROUP 12 - DATA VALUE 2 PARITY
```

```
2956  4AD9  FF          .BYTE   @377        ;DATA GROUP 12 - DATA VALUE 3
2957  4ADA  01          .BYTE   @001        ;DATA GROUP 12 - DATA VALUE 3 PARITY
2958  4ADB  FF          .BYTE   @377        ;DATA GROUP 12 - DATA VALUE 4
2959  4ADC  01          .BYTE   @001        ;DATA GROUP 12 - DATA VALUE 4 PARITY
2960
2961  4ADD  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 1
2962  4ADE  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 1 PARITY
2963  4ADF  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 2
2964  4AE0  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 2 PARITY
2965  4AE1  00          .BYTE   @000        ;DATA GROUP 13 - DATA VALUE 3
2966  4AE2  00          .BYTE   @000        ;DATA GROUP 13 - DATA VALUE 3 PARITY
2967  4AE3  00          .BYTE   @000        ;DATA GROUP 13 - DATA VALUE 4
2968  4AE4  00          .BYTE   @000        ;DATA GROUP 13 - DATA VALUE 4 PARITY
2969
2970  4AE5  FF          .BYTE   @377        ;DATA GROUP 14 - DATA VALUE 1
2971  4AE6  01          .BYTE   @001        ;DATA GROUP 14 - DATA VALUE 1 PARITY
2972  4AE7  FF          .BYTE   @377        ;DATA GROUP 14 - DATA VALUE 2
2973  4AE8  01          .BYTE   @001        ;DATA GROUP 14 - DATA VALUE 2 PARITY
2974  4AE9  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 3
2975  4AEA  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 3 PARITY
2976  4AEB  FF          .BYTE   @377        ;DATA GROUP 14 - DATA VALUE 4
2977  4AEC  01          .BYTE   @001        ;DATA GROUP 14 - DATA VALUE 4 PARITY
2978
2979  4AED  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 1
2980  4AEE  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 1 PARITY
2981  4AEF  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 2
2982  4AF0  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 2 PARITY
2983  4AF1  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 3
2984  4AF2  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 3 PARITY
2985  4AF3  00          .BYTE   @000        ;DATA GROUP 15 - DATA VALUE 4
2986  4AF4  00          .BYTE   @000        ;DATA GROUP 15 - DATA VALUE 4 PARITY
2987
2988  4AF5  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 1
2989  4AF6  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 1 PARITY
2990  4AF7  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 2
2991  4AF8  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 2 PARITY
2992  4AF9  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 3
2993  4AFA  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 3 PARITY
2994  4AFB  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 4
2995  4AFC  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 4 PARITY
2996
```

```
2998                                    .SBTTL  PROGRAM VARIABLES
2999
3000  4AFD   00                         UNITMP: .BYTE   0               ;UNIT MAP
3001  4AFE   00                         SUNIT:  .BYTE   0               ;PORT/TU SELECT BYTE
3002  4AFF   00   00                    AMTMSK: .WORD   0               ;AMTIE MASK WORD
3003  4B01   00   00                    INTBLTMP: .WORD 0               ;TEMP. STORAGE FOR THE INPUT TABLE POINTER
3004  4B03   00   00                    INTBL:  .WORD   0               ;INPUT TABLE ADDRESS STORAGE
3005  4B05   00   00                    OUTTBL: .WORD   0               ;OUTPUT TABLE ADDRESS STORAGE
3006  4B07   00                         DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
3007  4B08   00                         DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
3008  4B09   00                         DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
3009  4B0A   00                         DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
3010  4B0B   00                         SUBCNT: .BYTE   0               ;SUBGROUP COUNTER
3011         0000                               .END
```

```
A      -%0007          ADATA  = 0094          AMDCNA  4511  G       AMDCNB  4552  G
AMDCND  45D6  G        AMDCNE  4617  G        AMDCNG  468C  G       AMDCNH  46CD  G
AMDCNI  4769  G        AMDCNJ  4791  G        AMDCN1  438D  G       AMDCN2  43CE  G
AMDCN7  444B  G        AMDCN8  448C  G        AMDLPA  44DE          AMDLPB  4517
AMDLPC  4577           AMDLPD  45A4           AMDLPE  45DC          AMDLPG  4659
AMDLPH  4692           AMDLPI  471D           AMDLPJ  476F          AMDLP1  435F
AMDLP2  4393           AMDLP7  4418           AMDLP8  4451          AMDLP9  44B1
AMTIEP= 0001           AMTIE7= 0002           AMTMSK  4AFF          ARAIDF= 0098
ASAVE   4F9B           ATTCD   4F97           AXNUM   4F91          B      =%0000
BADST  = 0090          BIT0   = 0001          BIT1   = 0002         BIT15  = 8000
BIT2   = 0004          BIT3   = 0008          BIT4   = 0010         BIT5   = 0020
BIT6   = 0040          BIT7   = 0080          BIT8   = 0100         BIT9   = 0200
BRKPBC= 4F0A           BRKRAM= 4F10           BRKSTR= 4E60          BRKXCT= 4F00
BSAVE   4F9C           BYTCNT= 00D4           BYTEH   4F24          BYTEL   4F23
C      =%0001          CASCT   4F21           CASCTL= 00A0          CASSTA= 00A0
CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1          CBYTH  = 008B
CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086         CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094         CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028           CHOTIE= 0020          CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027           CKAMP   49D6  G       CKAM0   4913  G
CKAM1   492B  G        CKAM2   4943  G        CKAM3   495B  G       CKAM4   4973  G
CKAM5   498B  G        CKAM6   49A3  G        CKAM7   49BB  G       CKLOP  = 2815
CK5X4   48F8           CLEAR   48CD           CLKCTL= 00F0          CLOCK   4F26
CLRLP   48D1           CMC0H  = 0099          CMC0L  = 0098         CMC1H  = 009B
CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C         CMC3H  = 009F
CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096         CNTCTL= 00D7
CRCWRD= 0018           CSAVE   4F9D           CSRLH  = 0091         CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F         CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083         CXINL  = 0082
C.     - 0001          C.AVAI= 0080           C.DP   = 0008         C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080         C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040           C.WCS  = 0002         D      =%0002
DATAA   4B07           DATAAP  4B08           DATACT= 00D0          DATAE   4B09
DATAEP  4B0A           DBUS    4F28           DBUSCT= 00C0          DBUSST= 00C0
DDRA   = 00D8          DDRAIN= 0010           DDRB   = 00D9         DDRBIN= 0002
DDRC   = 00DA          DDRCIN= 0001           DDRCO  = 0088         DDRCTL= 00DB
DIAFLG  4F22           DIAGPG= 4300           DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010           DSAVE   4F9E          DSE    = 0006
DUMMX   4858  G        DUMMY   431E  G        D.ATH0= 0001          D.ATH1= 0002
D.EOTD- 0010           D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003         ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003           ERFLG   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812          ERLPE  = 280C         ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F           EXIT    48CA          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080          E.CRC  = 0080         E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001           E.TTEC= 0002          E.UNC  = 0004
FIFORD= 006A           FORMAT  4F25           FOUND   4344          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002           GOODTM= 0092          H      =%0004
HLSAVE  4FA0           IE     = 0008          INTBL   4B03          INTBLT  4B01
INTSTA= 00E0           ITERA   4F9A           I.PWR  = 0020         I.RMPE= 0040
```

```
I5.5   = 0010    I6.5   = 0020    I7.5   = 0040    KCALL  = 005F
KCLR   = 007B    KDEP   = 003F    KENAB  = 0078    KEXAM  = 003E
KEYBRD= 00C8    KEY1   = 0078    KEY10  = 006D    KEY11  = 006E
KEY12  = 006F    KEY13  = 005C    KEY14  = 005D    KEY15  = 005E
KEY16  = 005F    KEY17  = 003C    KEY18  = 003D    KEY19  = 003E
KEY2   = 0079    KEY20  = 003F    KEY3   = 007A    KEY4   = 007B
KEY5   = 0074    KEY6   = 0075    KEY7   = 0076    KEY8   = 0077
KEY9   = 006C    KINTA  = 006F    KLDAD  = 003D    KN0    = 003C
KN1    = 005C    KN2    = 005D    KN3    = 005E    KN4    = 006C
KN5    = 006D    KN6    = 006E    KN7    = 0074    KN8    = 0075
KN9    = 0076    KU2    = 0079    KU3    = 007A    KU8    = 0077
L      =%0005    LBLANK= 000F    LCE    = 000B    LCH    = 000C
LCL    = 000D    LCP    = 000E    LC0    = 0000    LC1    = 0001
LC2    = 0002    LC3    = 0003    LC4    = 0004    LC5    = 0005
LC6    = 0006    LC7    = 0007    LC8    = 0008    LC9    = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD  = 004C
LKKEY  = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR  = 0046    LPFLG   4F94
LPNUM   4F92    M      =%0006    MBSEL  = 00E0    MB.A   = 0008
MB.B   - 0004    MEMTOP= 4FFF    MINUS  = 000A    MM     = 8000
MSE    - 0008    MSGN   - 0017    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT= 0010    MT.Z   = 0008    M.ATA  = 0080    M.CAPE= 0020
M.CONT= 0080    M.DEM  = 0020    M.EBL  = 0010    M.EXC  = 0008
M.FAIL= 0008    M.ILR  - 0010    M.INIT= 0010    M.OCC  = 0020
M.ONLI= 0001    M.PE   = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN  = 0004    M.SCLK= 0001    M.TRA  = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5   = 0001    M6.5   = 0002    M7.5   = 0004    M8950I  49DD
M89500  4A7D    NOTCAP= 0088    OKAY   = 00FF    OPRRAM= 4300
OPSTRT= 0058    OPVER  = 0040    OUT     49DC    OUTTBL  4B05
PADCNT= 00D5    PADCRC= 0080    PDIAG  = 0048    PEID   = 008A
PENAB  = 004C    PESET  - 0001    PL     = 00B1    PRDD   = 004C
PRENF  - 009C    PS     - 00B2    PSTAT  = 0048    PSW    =%0009
P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080
P.LCS  = 0040    P.LWR  = 0020    P.RDP  = 0002    P.RPEN= 0004
P.RPST= 0002    P.RP0E= 0020    P.RP1E= 0010    P.RP2E= 0020
P.RP3E= 0010    P.SING= 0080    P.STAT= 0002    P.STPE= 0080
P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004    P.WDS  = 0040
P.WFLP= 0001    P.WPEN= 0010    P.WP0E= 0008    P.WP1E= 0004
P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002    QUE    = 281B
QUEM    281E    RAMT   = 0010    RARA   = 0006    RARAI  = 0004
RCHBD0  0048    RCHBD1= 0047    RCHOK  = 0046    RCHTST= 000C
RCLRT  = 000D    RCMD   = 0008    RCMLP  = 0003    RCONT  = 0080
RDATA  = 0017    RDCLK  = 0010    RDON   = 0011    READG  = 0007
REND   = 0014    REQST  = 2806    RESCHR= 00D1    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK  = 0002    RGCRI  = 0003
RIBG   = 0001    RILL   = 0012    RINST  = 000C    RMCTST= 0008
RMK2   = 0013    RNOP   = 0000    RPATH  = 0001    RPBAD  = 0044
RPCHI  - 0001    RPCLK  = 0003    RPCTL  = 0009    RPEI   = 0002
RPFAIL= 0000    RPF1   = 009D    RPF2   = 009E    RPOK   = 0043
```

```
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
R.DATA= 0040    R.DON = 0002    R.DRDY= 0010    R.END = 0010
R.ILL = 0004    R.JVOK= 0004    R.MK2 = 0008    R.PLOD= 0008
R.PLOO= 0010    R.PLO1= 0020    R.POST= 0020    R.STNM= 0002
R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080    R.TSTD= 0040
R.VOK = 0080    R00H  = 0081    R00L  = C080    R01H  = 0083
R01L  = 0082    R02H  = 0085    R02L  = 0084    R03H  = 0087
R03L  = 0086    R04H  = 0089    R04L  = 0038    R05H  = 008B
R05L  = 008A    R06H  = 008D    R06L  = 008C    R07H  = 008F
R07L  = 008E    R10H  = 0091    R10L  = 0090    R11H  = 0093
R11L  = 0092    R12H  = 0095    R12L  = 0094    R13H  = 0097
R13L  = 0096    R14H  = 0099    R14L  = 0098    R15H  = 009B
R15L  = 009A    R16H  = 009D    R16L  = 009C    R17H  = 009F
R17L  = 009E    R7.5  = 0010    SELCLR= 0000    SETATA= 00A1
SID   = 0080    SOD   = 0080    SOE   = 0040    SP    =%0008
SSCLK = 0040    SSTEP = 0005    STACK = 4FFF    STATRM= 4F20
STPCT   4F20    STRSP = 5000    SUBCNT  4B0B    SUNIT   4AFE
TADR00= C080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  - 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TEST2   4402    TEST3   44C8
TEST4   458E    TEST5   4643    TEST6   4709    TEST7   47CD
TMF   = 0099    TMRDY = 0040    TRKENA= 00D2    TSET  = 2803
TSTEND= 2818    TSTS  = 0040    TST01X  4351    TST02X  4407
TST03X  44CD    TST04X  4593    TST05X  4648    TST06X  470E
TST07A  4861    TST07B  4869    TST07C  4807    TST07X  47D2
TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010    T.ATH0= 0001
T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002    T.FPT = 0001
T.NTHR= 0004    T.ONL = 0020    T.PES = 0008    T.PSBJ= 0020
T.PSOJ- 0008    T.PS1J= 0010    T.RDY = 0080    T.RDYO= 0040
T.RWD = 0010    T.SCLK= 0002    UIBG  = 00A1    UNITMP  4AFD
UNITSL  4327    VALFC   4F98    VALTB   4F95    VELTST= 005B
WDR.P = 0010    WMCCTL- 00D3    WMCERR= 00DA    WMCSTA= 00D0
WRTCLK- 0000    WRTDAT= 00D3    W.ACRC= 0004    W.CRC = 0008
W.DIAG= 0002    W.DONN= 0040    W.ECC = 0010    W.ENAB= 0080
W.ERR = 0020    W.FMT = 0070    W.GCR = 0010    W.LEFT= 0004
W.ONES= 0020    W.RESI= 0002    W.REV = 0004    W.ROME= 0010
W.RST = 0001    W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020
X       %000A   X.DONN= 0080    X.ENAB- 0040    X.PEPE= 0002
X.ROME= 0001    X.WCLK- 0001    Y     =%000B    .     = 4B0C

ERRORS DETECTED:  0

*RPM3.A78/PTP,RPM3 NLIST,PARAM,MACRO,LIST,RPM3
RUN-TIME: 5 8 0 SECONDS
CORE USED:  10K
```

```
                                .TITLE  RPM4 - READ PATH MICRO CONTROLLER PART #4
                                .SBTTL  TEST 01 - INVALID 5X4 TRANSLATION TEST
                                ;ID     RPM4-READ PATH MICRO CONTROLLER PART #4
1332  4300                      ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*INVALID 5 X 4 TRANSLATION TEST
1334  4300                      SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
                                ;*AS WELL AS THE READ CHANNELS ABILITY TO DETECT ALL INVALID 5 TO 4
                                ;*TRANSLATIONS POSSIBLE.
1338  4300                      SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*    DELAY 50 MICROSECONDS
                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*    CLEAR THE TIE BUS
                                ;*    SET UP INPUT TABLE POINTER
                                ;*  BGNDO
                                ;*  : ENABLE THE READ PATH CLOCK
                                ;*  : SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
                                ;*  : CLOCK THE FIFO
                                ;*  : ISSUE CLEAR ALL COMMAND
                                ;*  : WAIT
                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*  : INIT THE LOOP COUNT TO 5
                                ;*  : BGNDO
                                ;*  : :    MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
                                ;*  : :    INCREMENT THE INPUT TABLE POINTER
                                ;*  : :    MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
                                ;*  : :    CLOCK THE DATA INTO THE FIFO
                                ;*  : :    DECREMENT THE LOOP COUNT
                                ;*  : :    INCREMENT THE INPUT TABLE POINTER
                                ;*  : :    DO UNTIL LOOP COUNT - 0
                                ;*  : ENDDO
                                ;*  : ISSUE DIAGNOSTIC READ COMMAND
                                ;*  : SET UP WATCHDOG TIMER COUNT
                                ;*  : BGNDO
                                ;*  : :    SINGLE STEP THE READ PATH
                                ;*  : :    DECREMENT THE WATCHDOG TIMER
                                ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
                                ;*  : ENDDO
                                ;*  : IF WATCHDOG TIMER 0
                                ;*  : :    THEN-ERROR
```

```
1374                                ;*  : :    ELSE-CONTINUE
1375                                ;*  : ENDIF
1376                                ;*  : SINGLE STEP THE READ PATH
1377                                ;*  : CALL SUBROUTINE CKILL
1378                                ;*  : DO UNTIL END OF INPUT DATA TABLE
1379                                ;*  ENDDO
1380                                ;*ENDTST
1381  4300                          SE
 (1)                                ; ************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1382                                ;*RPM4 MICRO TEST 01
1383                                ;*RPM4 MICRO ERROR 01
1384                                ;*RPM4-READ PATH-INVALID 5X4 TRANSLATION TEST
1385                                ;*M8950'S, M8953
1386                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1387                                ;*FATAL ERROR - TEST ABORTED
1388                                ;*
1389                                ;*RPM4 MICRO TEST 01
1390                                ;*RPM4 MICRO ERROR 03
1391                                ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1392                                ;*M8950'S, M8953
1393                                ;*CHANNEL 0 NOT ILLEGAL 5X4
1394                                ;*
1395                                ;*RPM4 MICRO TEST 01
1396                                ;*RPM4 MICRO ERROR 04
1397                                ;*RPM4-READ PATH-INVALID 4X4 TRANS_ATION TEST
1398                                ;*M8950'S, M8953
1399                                ;*CHANNEL 1 NOT ILLEGAL 5X4
1400                                ;*
1401                                ;*RPM4 MICRO TEST 01
1402                                ;*RPM4 MICRO ERROR 05
1403                                ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1404                                ;*M8950'S, M8953
1405                                ;*CHANNEL 2 NOT ILLEGAL 5X4
1406                                ;*
1407                                ;*RPM4 MICRO TEST 01
1408                                ;*RPM4 MICRO ERROR 06
1409                                ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1410                                ;*M8950'S, M8953
1411                                ;*CHANNEL 3 NOT ILLEGAL 5X4
1412                                ;*
1413                                ;*RPM4 MICRO TEST 01
1414                                ;*RPM4 MICRO ERROR 07
1415                                ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1416                                ;*M8950'S, M8953
1417                                ;*CHANNEL 4 NOT ILLEGAL 5X4
1418                                ;*
1419                                ;*RPM4 MICRO TEST 01
1420                                ;*RPM4 MICRO ERROR 10
1421                                ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1422                                ;*M8950'S, M8953
1423                                ;*CHANNEL 5 NOT ILLEGAL 5X4
1424                                ;*
```

```
                                        ;*RPM4 MICRO TEST 01
                                        ;*RPM4 MICRO ERROR 11
                                        ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 6 NOT ILLEGAL 5X4
                                        ;*
                                        ;*RPM4 MICRO TEST 01
                                        ;*RPM4 MICRO ERROR 12
                                        ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 7 NOT ILLEGAL 5X4
                                        ;*
                                        ;*RPM4 MICRO TEST 01
                                        ;*RPM4 MICRO ERROR 13
                                        ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P NOT ILLEGAL 5X4
1442 4300                               $
 (1)                                    ; ************************************************************************
1443
1444 4300                               TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1) 4300 3E 01              7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302 CD 03 28          18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%RPM4-READ PATH-INVALID 5X4 TRANSLATION TEST
                                        ;&M8950'S, M8953
1447 4305                                       REQ     @7,0,0,0,0
 (1) 4305 CD 06 28          18.0                 CALL    REQST
 (1) 4308 00                                     .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4309 00 00                                  .WORD   0               ;SYSTEM "0" COUNT
 (1) 430B 00 00                                  .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D 00                                     .BYTE   0               ;DATA COMPARE FLAG IF  1
 (1) 430E 07                                     .BYTE   @7              ;REQUEST CODE
1448 430F                                       RIN     R12L
 (1) 430F DB 94             10.0                 IN      R12L            ;READ R12L INTO AC
 (1) 4311 7F                 4.0                 MOV     A,A             ;RETRY LINK
1449 4312 32 3F 48          13.0                RTA     UNITMP
1450 4315 A7                 4.0                ANA     A
1451 4316 C2 27 43          10.0                JNZ     UNITSL
1452 4319                                       ERR     EXIT,DUMMY
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 4319 CD 09 28          18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      0001                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 431C 01                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 431D 00                                     .BYTE
 (1) 431E CD 15 28          18.0         DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4321 DA DD 44          10.0                 JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1453                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1454                                    ;>FATAL ERROR - TEST ABORTED
1455 4324 C3 DD 44          10.0                JMP     EXIT
1456
1457 4327 06 00              7.0        UNITSL: MVI     B,@0            ;LOAD THE BASE UNIT NUMBER
1458 4329 3A 3F 48          13.0                LDA     UNITMP          ;DID THE USER SPECIFY TU
1459 432C E6 01              7.0                ANI     @01             ;PORT 0?
1460 432E C2 44 43          10.0                JNZ     FOUND           ;YES-GO USE PORT #0
```

```
1461  4331  04          4.0          INR   B          ;NO-UPDATE POINTER TO PORT #1
1462  4332  3A 3F 48   13.0          LDA   UNITMP     ;DID THE USER SPECIFY TU
1463  4335  E6 02      7.0          ANI   @02        ;PORT 1?
1464  4337  C2 44 43   10.0          JNZ   FOUND      ;YES-GO USE PORT #1
1465  433A  04          4.0          INR   B          ;NO-UPDATE POINTER TO PORT #2
1466  433B  3A 3F 48   13.0          LDA   UNITMP     ;DID THE USER SPECIFY TU
1467  433E  E6 04      7.0          ANI   @04        ;PORT 2?
1468  4340  C2 44 43   10.0          JNZ   FOUND      ;YES-GO USE PORT #2
1469  4343  04          4.0          INR   B          ;NO-ASSUME PORT #3
1470  4344  CD E0 44   18.0  FOUND:   CALL  CLEAR      ;CLEAR ALL THE TU PORTS
1471  4347  DB E0      10.0          IN    INTSTA     ;GET THE MASS BUS SELECT BIT
1472  4349  E6 80      7.0          ANI   BIT7       ;
1473  434B  B0          4.0          ORA   B          ;OR IN THE TU PORT SELECT BITS
1474  434C  D3 E0      10.0          OUT   MBSEL      ;SELECT THE MASS BUS AND TU PORTS
1475  434E  32 40 48   13.0          STA   SUNIT      ;STORE THE UNIT NUMBER FOR LATER USE
1476
1477  4351  3E 10      7.0  TST01X: MVI   A,RDCLK    ;SET NORMAL READ PATH CLOCKS
1478  4353  D3 F0      10.0          OUT   CLKCTL
1479
1480  4355  3E 04      7.0          MVI   A,P.RPEN           ;ENABLE THE READ PATH FROM THE TU PORT
1481  4357  D3 4C      10.0          OUT   PENAB      ;
1482  4359  3E 10      7.0          MVI   A,W.GCR    ;SET GCR MODE
1483  435B  D3 D3      10.0          OUT   WMCCTL
1484  435D  3E 60      7.0          MVI   A,P.LWR!P.LCS  ;SET THE PORT CONTROL TO
1485  435F  D3 48      10.0          OUT   PDIAG      ;LCS, LWR AND DPEN
1486  4361  AF          4.0          XRA   A
1487  4362  D3 44      10.0          OUT   TAMT       ;
1488  4364  3E 0A      7.0          MVI   A,10       ;DELAY
1489  4366  3D          4.0  1$:     DCR   A          ;50
1490  4367  C2 66 43   10.0          JNZ   1$         ;MICROSECONDS
1491
1492  436A  D3 D2      10.0          OUT   TRKENA     ;CLEAR ALL TRACKS FROM
1493  436C  D3 D2      10.0          OUT   TRKENA     ;THE TRANSLATOR
1494
1495  436E  AF          4.0          XRA   A          ;CLEAR THE TIE BUS
1496  436F  D3 0A      10.0          OUT   RTIEB
1497
1498  4371  21 93 46   10.0          LXI   H,M8950E   ;SET UP POINTER TO INPUT DATA TABLE
1499  4374  22 45 48   16.0          SHLD  INTBL      ;SAVE IT
1500  4377  3E A8      7.0  TST01C: MVI   A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1501                                                 ;TIE BUS JAM AND PLO DISABLE
1502  4379  D3 09      10.0          OUT   RPCTL
1503  437B  D3 08      10.0          OUT   RFIFOL     ;CLOCK FIFO TO CLEAR GCR DATA FF'S ON M8950'S
1504  437D  3E 0D      7.0          MVI   A,RCLRT    ;ISSUE CLEAR ALL COMMAND
1505  437F  D3 08      10.0          OUT   RCMD
1506
1507  4381  00          4.0          NOP              ;WAIT
1508  4382  00          4.0          NOP
1509  4383  00          4.0          NOP
1510  4384  00          4.0          NOP
1511
1512  4385  3E A9      7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1513                                                                      ;SET TIE BUS JAM AND PLO DISABLE
1514  4387  D3 09      10.0          OUT   RPCTL
```

```
1515
1516  4389  0E  02        7.0          MVI   C,2             ;SET UP THE FIFO LOAD COUNTER
1517  438B  2A  45  48    16.0  3$:    LHLD  INTBL           ;GET POINTER TO INPUT DATA TABLE
1518  438E  06  05         7.0          MVI   B,@5            ;SET UP LOOP COUNT
1519  4390  7E             7.0  1$:     MOV   A,M             ;GET A DATA BYTE
1520  4391  D3  40        10.0          OUT   TCMD            ;STORE DATA IN COMMAND ADDRESS
1521  4393  23             6.0          INX   H               ;POINT TO DATA PARITY
1522  4394  7E             7.0          MOV   A,M             ;GET THE DATA PARITY
1523  4395  07             4.0          RLC                   ;POSITION FOR OUTPUT
1524  4396  F6  60         7.0          ORI   P.LWR.P.LCS  ;OR IN CONTROL BITS
1525  4398  D3  48        10.0          OUT   PDIAG           ;OUTPUT THE DATA PARITY
1526  439A  D3  08        10.0          OUT   RFIFOL          ;CLOCK DATA INTO THE FIFO'S
1527
1528  439C  23             6.0          INX   H               ;UPDATE THE TABLE POINTER
1529  439D  05             4.0          DCR   B               ;DECREMENT LOOP COUNT
1530  439E  C2  90  43    10.0          JNZ   1$              ;DO UNTIL LOOP COUNT = 0
1531
1532  43A1  0D             4.0          DCR   C               ;DECREMENT THE FIFO LOAD LOOP COUNTER
1533  43A2  C2  88  43    10.0          JNZ   3$              ;CONTINUE UNTIL ZERO
1534                                ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1535
1536  43A5  22  45  48    16.0          SHLD  INTBL           ;SAVE THE INPUT DATA TABLE POINTER
1537  43A8  3E  0B         7.0          MVI   A,DIARD         ;LOAD THE DIAGNOSTIC READ
1538  43AA  D3  0B        10.0          OUT   RCMD            ;COMMAND
1539  43AC  11  01  00    10.0          LXI   D,1             ;SET WATCH DOG INCREMENT
1540  43AF  21  A8  FD    10.0          LXI   H,-600          ;SET WATCH DOG COUNT TO 600
1541  43B2  D3  0C        10.0  2$:     OUT   RINST           ;STEP THE READ PATH
1542  43B4  19            10.0          DAD   D               ;WATCH DOG TIMEOUT?
1543  43B5  D2  B2  43    10.0          JNC   2$              ;NO-CONTINUE
1544  43B8  D3  0C        10.0  TST01A: OUT   RINST           ;SINGLE STEP THE READ PATH
1545  43BA  CD  0B  45    18.0          CALL  CKILL           ;CHECK FOR FAILING READ CHANNELS
1546  43BD  DA  51  43    10.0          JC    TST01X          ;LOOP ON ERROR
1547  43C0  2A  45  48    16.0          LHLD  INTBL           ;GET THE INPUT TABLE POINTER
1548  43C3  2B             6.0          DCX   H               ;UPDATE TO LAST PARITY BYTE
1549  43C4  7E             7.0          MOV   A,M             ;GET LAST PARITY BYTE
1550  43C5  A7             4.0          ANA   A               ;SET CONDITION BITS
1551  43C6  F2  77  43    10.0          JP    TST01C          ;CONTINUE TO END INPUT TABLE
1552  43C9  CD  E0  44    18.0          CALL  CLEAR           ;CLEAR ALL TU PORTS
1553  43CC                              ENDTST TST01X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43CC                              REQ   7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43CC  CD  06  28    18.0                        CALL    REQST
 (2)  43CF  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  43D0  00  00                                    .WORD                   ;SYSTEM '*' COUNT
 (2)  43D2  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43D4  00                                        .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  43D5  07                                        .BYTE   7               ;REQUEST CODE
 (1)  43D6  3A  9A  4F    13.0          LDA   ITERA           ;GET ITERATION COUNT
 (1)  43D9  3D             4.0          DCR   A               ;DOWNCOUNT
 (1)  43DA  32  9A  4F    13.0          STA   ITERA           ;SAVE COUNT
 (1)  43DD  F2  51  43    10.0          JP    TST01X          ;DO TEST UNTIL TILL = 0
1554
```

```
                                .SBTTL  TEST 02 - 5 X 4 TRANSLATION TEST (REVERSE)
                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*5 X 4 TRANSLATION TEST (REVERSE)
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
                                ;*AS WELL AS THE READ CHANNELS ABILITY TO 5 TO 4 TRANSLATE ALL POSSIBLE
                                ;*COMBINATIONS OF INPUT DATA IN THE REVERSE MODE.
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*     DELAY 50 MICROSECONDS
                                ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*     CLEAR THE TIE BUS
                                ;*     SET UP INPUT TABLE POINTER
                                ;*     SET UP OUTPUT TABLE POINTER
                                ;*   BGNDO
                                ;*   : ENABLE THE READ PATH CLOCK
                                ;*   : SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
                                ;*   : CLOCK THE FIFO
                                ;*   : ISSUE CLEAR ALL COMMAND
                                ;*   : WAIT
                                ;*   : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*   : PUT THE CONTROLLER IN GCR REVERSE MODE
                                ;*   : INIT THE SUB-GROUP COUNT TO 2
                                ;*   : BGNDO
                                ;*   : : INIT THE LOOP COUNT TO 5
                                ;*   : : BGNDO
                                ;*   : : :     MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
                                ;*   : : :     INCREMENT THE INPUT TABLE POINTER
                                ;*   : : :     MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
                                ;*   : : :     CLOCK THE DATA INTO THE FIFO
                                ;*   : : :     DECREMENT THE LOOP COUNT
                                ;*   : : :     INCREMENT THE INPUT TABLE POINTER
                                ;*   : : :     DO UNTIL LOOP COUNT = 0
                                ;*   : : ENDDO
                                ;*   : : DECREMENT THE SUB-GROUP COUNTER
                                ;*   : : DO UNTIL THE SUB-GROUP COUNTER -0
                                ;*   : ENDDO
                                ;*   : ISSUE DIAGNOSTIC READ COMMAND
                                ;*   : SET UP WATCHDOG TIMER COUNT
                                ;*   : BGNDO
                                ;*   : :     SINGLE STEP THE READ PATH
```

```
1601                                    ;*   : :    DECREMENT THE WATCHDOG TIMER
1602                                    ;*   : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1603                                    ;*   : ENDDO
1604                                    ;*   : IF WATCHDOG TIMER=0
1605                                    ;*   : :    THEN-ERROR
1606                                    ;*   : :    ELSE-CONTINUE
1607                                    ;*   : ENDIF
1608                                    ;*   : SINGLE STEP THE READ PATH
1609                                    ;*   : INIT THE LOOP COUNT TO 4
1610                                    ;*   : BGNDO
1611                                    ;*   : :    SINGLE STEP THE READ PATH
1612                                    ;*   : :    INPUT THE ACTUAL DATA
1613                                    ;*   : :    INPUT THE ACTUAL PARITY
1614                                    ;*   : :    GET THE EXPECTED DATA FROM THE EXPECTED DATA TABLE
1615                                    ;*   : :    INCREMENT THE OUTPUT TABLE POINTER
1616                                    ;*   : :    GET THE EXPECTED PARITY FROM THE EXPECTED DATA TABLE
1617                                    ;*   : :    INCREMENT THE OUTPUT TABLE POINTER
1618                                    ;*   : :    CALL SUBROUTINE ''CK5X4''
1619                                    ;*   : :    DECREMENT THE LOOP COUNT
1620                                    ;*   : :    DO UNTIL LOOP COUNT = 0
1621                                    ;*   : ENDDO
1622                                    ;*   :  DO UNTIL END OF INPUT DATA TABLE
1623                                    ;*  ENDDO
1624                                    ;*ENDTST
1625  43E0                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1626                                    ;*RPM4 MICRO TEST 02
1627                                    ;*RPM4 MICRO ERROR 02
1628                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1629                                    ;*M8950'S, M8953
1630                                    ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1631                                    ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1632                                    ;*FATAL ERROR - MICRO TEST ABORTED
1633                                    ;*
1634                                    ;*RPM4 MICRO TEST 02
1635                                    ;*RPM4 MICRO ERROR 14
1636                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1637                                    ;*M8950'S, M8953
1638                                    ;*CHANNEL 0 FAILED
1639                                    ;*ACTUAL = NNNN
1640                                    ;*EXPECTED = NNNN
1641                                    ;*SUBGROUP NUMBER = LLL
1642                                    ;*
1643                                    ;*RPM4 MICRO TEST 02
1644                                    ;*RPM4 MICRO ERROR 15
1645                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1646                                    ;*M8950'S, M8953
1647                                    ;*CHANNEL 1 FAILED
1648                                    ;*ACTUAL = NNNN
1649                                    ;*EXPECTED = NNNN
1650                                    ;*SUBGROUP NUMBER = LLL
1651                                    ;*
```

```
1652                    ;*RPM4 MICRO TEST 02
1653                    ;*RPM4 MICRO ERROR 16
1654                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1655                    ;*M8950'S, M8953
1656                    ;*CHANNEL 2 FAILED
1657                    ;*ACTUAL = NNNN
1658                    ;*EXPECTED = NNNN
1659                    ;*SUBGROUP NUMBER = LLL
1660                    ;*
1661                    ;*RPM4 MICRO TEST 02
1662                    ;*RPM4 MICRO ERROR 17
1663                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1664                    ;*M8950'S, M8953
1665                    ;*CHANNEL 3 FAILED
1666                    ;*ACTUAL = NNNN
1667                    ;*EXPECTED = NNNN
1668                    ;*SUBGROUP NUMBER = LLL
1669                    ;*
1670                    ;*RPM4 MICRO TEST 02
1671                    ;*RPM4 MICRO ERROR 20
1672                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1673                    ;*M8950'S, M8953
1674                    ;*CHANNEL 4 FAILED
1675                    ;*ACTUAL = NNNN
1676                    ;*EXPECTED = NNNN
1677                    ;*SUBGROUP NUMBER = LLL
1678                    ;*
1679                    ;*RPM4 MICRO TEST 02
1680                    ;*RPM4 MICRO ERROR 21
1681                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1682                    ;*M8950'S, M8953
1683                    ;*CHANNEL 5 FAILED
1684                    ;*ACTUAL = NNNN
1685                    ;*EXPECTED = NNNN
1686                    ;*SUBGROUP NUMBER = LLL
1687                    ;*
1688                    ;*RPM4 MICRO TEST 02
1689                    ;*RPM4 MICRO ERROR 22
1690                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1691                    ;*M8950'S, M8953
1692                    ;*CHANNEL 6 FAILED
1693                    ;*ACTUAL = NNNN
1694                    ;*EXPECTED = NNNN
1695                    ;*SUBGROUP NUMBER = LLL
1696                    ;*
1697                    ;*RPM4 MICRO TEST 02
1698                    ;*RPM4 MICRO ERROR 23
1699                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1700                    ;*M8950'S, M8953
1701                    ;*CHANNEL 7 FAILED
1702                    ;*ACTUAL = NNNN
1703                    ;*EXPECTED = NNNN
1704                    ;*SUBGROUP NUMBER = LLL
1705                    ;*
```

```
1706                                    ;*RPM4 MICRO TEST 02
1707                                    ;*RPM4 MICRO ERROR 24
1708                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1709                                    ;*M8950'S, M8953
1710                                    ;*CHANNEL P FAILED
1711                                    ;*ACTUAL = NNNN
1712                                    ;*EXPECTED = NNNN
1713                                    ;*SUBGROUP NUMBER = LLL
1714  43E0                              S
 (1)                                    ; ***************************************************************************************
1715
1716  43E0                              TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  43E0  3E  02            7.0                MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  43E2  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1717                                    ;%RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1718                                    ;&M8950'S, M8953
1719
1720  43E5  3E  10               7.0    TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1721  43E7  D3  F0              10.0            OUT     CLKCTL
1722
1723  43E9  CD  E0  44          18.0            CALL    CLEAR           ;CLEAR ALL TU PORTS
1724  43EC  3A  40  48          13.0            LDA     SUNIT
1725  43EF  D3  E0              10.0            OUT     MBSEL
1726  43F1  3E  04               7.0            MVI     A,P.RPEN
1727  43F3  D3  4C              10.0            OUT     PENAB
1728
1729  43F5  3E  60               7.0            MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1730  43F7  D3  48              10.0            OUT     PDIAG           ;LCS, LWR AND DPEN
1731  43F9  AF                   4.0            XRA     A
1732  43FA  D3  44              10.0            OUT     TAMT
1733  43FC  3E  0A               7.0            MVI     A,10            ;DELAY
1734  43FE  3D                   4.0    1$:     DCR     A               ;50
1735  43FF  C2  FE  43          10.0            JNZ     1$              ;MICROSECONDS
1736
1737  4402  D3  D2              10.0            OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1738  4404  D3  D2              10.0            OUT     TRKENA          ;THE TRANSLATOR
1739
1740  4406  AF                   4.0            XRA     A               ;CLEAR THE TIE BUS
1741  4407  D3  0A              10.0            OUT     RTIEB
1742  4409  3E  01               7.0            MVI     A,@1            ;INITIALIZE THE SUBGROUP COUNT
1743  440B  32  4D  48          13.0            STA     SUBCNT          ;
1744
1745  440E  21  1F  47          10.0            LXI     H,M8950I        ;SET UP POINTER TO INPUT DATA TABLE
1746  4411  22  45  48          16.0            SHLD    INTBL           ;SAVE IT
1747  4414  21  BF  47          10.0            LXI     H,M89500        ;SET UP POINTER TO OUTPUT DATA TABLE
1748  4417  22  47  48          16.0            SHLD    OUTTBL          ;SAVE IT
1749  441A  3E  A8               7.0    TST02C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1750                                                                    ;TIE BUS JAM AND PLO DISABLE
1751  441C  D3  09              10.0            OUT     RPCTL
1752  441E  3E  14               7.0            MVI     A,W.GCR!W.REV           ;SET GCR MODE
1753  4420  D3  D3              10.0            OUT     WMCCTL
1754  4422  D3  08              10.0            OUT     RFIFOL          ;CLOCK FIFO TO CLEAR GCR DATA FF'S ON M8950'S
1755  4424  3E  0D               7.0            MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1756  4426  D3  0B              10.0            OUT     RCMD
```

```
1757  4428  00                  4.0          NOP                          ;WAIT
1758  4429  00                  4.0          NOP
1759  442A  00                  4.0          NOP
1760  442B  00                  4.0          NOP
1761  442C  3E  A9              7.0          MVI     A,R.PLO1.R.STPC.R.PLOD!R.TBJN    ;STOP THE READ PATH
1762                                                                                   ;SET TIE BUS JAM AND PLO DISABLE
1763  442E  D3  09             10.0          OUT     RPCTL
1764
1765  4430  0E  02              7.0          MVI     C,2              ;INITIALIZE THE SUB-GROUP COUNTER
1766  4432  2A  45  48          16.0  3$:     LHLD    INTBL            ;GET POINTER TO INPUT DATA TABLE
1767  4435  06  05              7.0          MVI     B,@5             ;SET UP LOOP COUNT
1768  4437  7E                  7.0  1$:     MOV     A,M              ;GET A DATA BYTE
1769  4438  D3  40             10.0          OUT     TCMD             ;STORE DATA IN COMMAND ADDRESS
1770  443A  23                  6.0          INX     H                ;POINT TO DATA PARITY
1771  443B  7E                  7.0          MOV     A,M              ;GET THE DATA PARITY
1772
1773  443C  07                  4.0          RLC                      ;POSITION FOR OUTPUT
1774  443D  F6  60              7.0          ORI     P.LWR!P.LCS      ;OR IN CONTROL BITS
1775
1776  443F  D3  48             10.0          OUT     PDIAG            ;OUTPUT THE DATA PARITY
1777  4441  D3  08             10.0          OUT     RFIFOL           ;CLOCK DATA INTO THE FIFO'S
1778  4443  23                  6.0          INX     H                ;UPDATE THE TABLE POINTER
1779  4444  05                  4.0          DCR     B                ;DECREMENT LOOP COUNT
1780  4445  C2  37  44          10.0          JNZ     1$               ;DO UNTIL LOOP COUNT = 0
1781  4448  0D                  4.0          DCR     C                ;DECREMENT THE SUBGROUP COUNTER
1782  4449  C2  32  44          10.0          JNZ     3$               ;LOAD THE SUB-GROUP TWICE
1783                                  ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1784  444C  22  43  48          16.0          SHLD    INTBLTMP         ;SAVE THE INPUT DATA TABLE POINTER
1785  444F  3E  0B              7.0          MVI     A,DIARD          ;LOAD THE DIAGNOSTIC READ
1786  4451  D3  0B             10.0          OUT     RCMD             ;COMMAND
1787  4453  11  01  00          10.0          LXI     D,1              ;SET WATCH DOG INCREMENT
1788  4456  21  A8  FD          10.0          LXI     H,-600           ;SET WATCH DOG COUNT TO 600
1789  4459  D3  0C             10.0  2$:     OUT     RINST            ;STEP THE READ PATH
1790  445B  DB  01             10.0          IN      RPCHI            ;DATA READY SET?
1791  445D  E6  10              7.0          ANI     R.DRDY
1792  445F  C2  74  44          10.0          JNZ     TST02A           ;YES-GO PROCESS
1793  4462  19                  10.0          DAD     D                ;WATCH DOG TIMEOUT?
1794  4463  D2  59  44          10.0          JNC     2$               ;NO-CONTINUE
1795  4466                                    ERR     EXIT,DUMMX
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4466  CD  09  28          18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0002                               MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4469  02                               .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  446A  00                               .BYTE
 (1)  446B  CD  15  28          18.0  DUMMX:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  446E  DA  DD  44          10.0          JC      EXIT             ;LOOP ADDRESS IF LOOP SPECIFIED
1796                            ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1797                            ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1798                            ;<FATAL ERROR - MICRO TEST ABORTED
1799  4471  C3  DD  44          10.0          JMP     EXIT
1800
1801  4474  2A  47  48          16.0  TST02A: LHLD    OUTTBL           ;GET ADDRESS OF EXPECTED OUTPUT DATA TABLE
1802  4477  0C                   4 0          NOP
1803  4478  00                   4.0          NOP
```

```
1804  4479  00              4.0          NOP
1805  447A  0E  04          7.0          MVI     C,@4            ;INIT THE LOOP COUNT
1806  447C  00              4.0  TST02B: NOP
1807  447D  D3  0C          10.0         OUT     RINST           ;SINGLE STEP THE READ PATH
1808  447F  DB  17          10.0         IN      RDATA           ;INPUT THE ACTUAL DATA BYTE
1809  4481  32  49  48      13.0         STA     DATAA           ;STORE IT
1810  4484  DB  15          10.0         IN      RPSTA           ;INPUT THE ACTUAL PARITY BIT
1811  4486  E6  40          7.0          ANI     R.DATA          ;REMOVE JUNK BITS
1812  4488  07              4.0          RLC                     ;POSITION THE BIT FOR COMPARE
1813  4489  07              4.0          RLC
1814  448A  32  4A  48      13.0         STA     DATAAP          ;STORE IT
1815  448D  7E              7.0          MOV     A,M             ;GET THE EXPECTED DATA BYTE
1816  448E  32  4B  48      13.0         STA     DATAE           ;STORE IT
1817  4491  23              6.0          INX     H               ;UPDATE THE POINTER
1818  4492  7E              7.0          MOV     A,M             ;GET THE EXPECTED PARITY BYTE
1819  4493  32  4C  48      13.0         STA     DATAEP          ;STORE IT
1820  4496  EB              4.0          XCHG                    ;SAVE H AND L IN D AND E
1821  4497  3A  4D  48      13.0         LDA     SUBCNT          ;GET THE SUBGROUP COUNT
1822  449A                               ROUT    R05L            ;STORE IN CAS
 (1)  449A  D3  8A          10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  449C  7F              4.0                  MOV     A,A             ;RETRY LINK
1823  449D  AF              4.0          XRA     A               ;CLEAR THE HIGH HALF OF THE SUBGROUP COUNTER
1824  449E                               ROUT    R05H            ;
 (1)  449E  D3  8B          10.0                 OUT     R05H            ;WRITE AC INTO R05H
 (1)  44A0  7F              4.0                  MOV     A,A             ;RETRY LINK
1825  44A1  CD  AE  45      18.0         CALL    CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
1826  44A4  DA  1A  44      10.0         JC      TST02C          ;IF LOOP ON ERROR REPEAT THE TEST
1827  44A7  EB              4.0          XCHG                    ;RESTORE H AND L FROM D AND E
1828  44A8  23              6.0          INX     H               ;UPDATE THE EXPECTED DATA TABLE POINTER
1829  44A9  0D              4.0          DCR     C               ;DECREMENT THE LOOP COUNT
1830  44AA  C2  7C  44      10.0         JNZ     TST02B          ;DO UNTIL LOOP COUNT - ZERO
1831  44AD  22  47  48      16.0         SHLD    OUTTBL          ;SAVE EXPECTED DATA TABLE POINTER
1832  44B0  3A  4D  48      13.0         LDA     SUBCNT          ;GET THE SUBGROUP COUNTER
1833  44B3  3C              4.0          INR     A               ;UPDATE FOR THE NEXT SUBGROUP
1834  44B4  32  4D  48      13.0         STA     SUBCNT          ;SAVE IT
1835  44B7  2A  43  48      16.0         LHLD    INTBLTMP        ;GET THE INPUT TABLE POINTER
1836  44BA  22  45  48      16.0         SHLD    INTBL           ;UPDATE THE INPUT TABLE POINTER
1837  44BD  2B              6.0          DCX     H               ;UPDATE TO LAST PARITY BYTE
1838  44BE  7E              7.0          MOV     A,M             ;GET LAST PARITY BYTE
1839  44BF  A7              4.0          ANA     A               ;SET CONDITION BITS
1840  44C0  F2  1A  44      10.0         JP      TST02C          ;CONTINUE UNTIL END OF INPUT TABLE
1841  44C3  CD  E0  44      18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
```

```
1843  44C6                                        ENDTST  TST02X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44C6                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44C6  CD  06  28        18.0                                  CALL    REQST
 (2)  44C9  00                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  44CA  00  00                                                  .WORD                   ;SYSTEM '*' COUNT
 (2)  44CC  00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44CE  00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44CF  07                                                      .BYTE   7               ;REQUEST CODE
 (1)  44D0  3A  9A  4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  44D3  3D                 4.0            DCR     A               ;DOWNCOUNT
 (1)  44D4  32  9A  4F        13.0            STA     ITERA           ;SAVE COUNT
 (1)  44D7  F2  E5  43        10.0            JP      TST02X          ;DO TEST UNTIL TILL   0
1844  44DA  CD  E0  44        18.0            CALL    CLEAR
1845
1846  44DD  C3  18  28        10.0  EXIT:     JMP     TSTEND
```

```
1848                                    .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
1849  44E0                              SSUB
 (1)                                    ;*********************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
1850                                    ;*CLEAR ALL TU PORTS
1851  44E0                              SD
 (1)                                    ;*********************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1852                                    ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
1853                                    ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
1854                                    ;*AND LOOP MODES.
1855  44E0                              SP
 (1)                                    ;*********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1856                                    ;*BGNSUB
1857                                    ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
1858                                    ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
1859                                    ;*     CLEAR PORT SELECT FOR TRANSPORT
1860                                    ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
1861                                    ;*     CLEAR PORT DIAGNOSTIC CONTROL
1862                                    ;*     CLEAR PORT AMTIE WORD
1863                                    ;*ENDSUB
1864  44E0                              S
 (1)                                    ;*********************************************************************
1865  44E0  F5                  12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
1866  44E1  C5                  12.0          PUSH    B
1867  44E2  06  00               7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
1868  44E4  DB  E0              10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
1869  44E6  E6  80                7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
1870  44E8  B0                    4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1871  44E9  D3  E0              10.0          OUT     MBSEL           ;RESET TO THIS PORT
1872  44EB  3E  80                7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
1873  44ED  D3  40              10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
1874  44EF  AF                    4.0          XRA     A               ;CLEAR TU COMMAND A
1875  44F0  D3  40              10.0          OUT     TCMD            ;
1876  44F2  3E  81                7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
1877  44F4  D3  40              10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
1878  44F6  3E  00                7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
1879  44F8  D3  40              10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
1880  44FA  AF                    4.0          XRA     A
1881  44FB  D3  44              10.0          OUT     TAMT            ;CLEAR AMTIE WORD
1882  44FD  D3  48              10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
1883  44FF  D3  4C              10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
1884  4501  04                    4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
1885  4502  78                    4.0          MOV     A,B
1886  4503  FE  04                7.0          CPI     4               ;DONE?
1887  4505  C2  E4  44           10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
1888  4508  C1                   10.0          POP     B               ;RESET B & C
1889  4509  F1                   10.0          POP     PSW             ;ALL DONE
1890  450A  C9                   10.0          RET                     ;EXIT
```

```
1892                                .SBTTL  SUBROUTINE CHECK ILLEGAL 5X4
1893  4508                          SSUB
 (1)                                ;*****************************************************************************
 (1)                                ;*SUBROUTINE TITLE
 (1)                                ;*---------- -----
1894                                ;*CHECK READ CHANNELS ILLEGAL 5X4
1895  4508                          SD
 (1)                                ;*****************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1896                                ;*THIS SUBROUTINE IS USED TO CHECK THAT ALL READ CHANNEL ''ILLEGAL 5X4''
1897                                ;*BITS EQUAL ONE AND TO FLAG ANY EXCEPTIONS WITH AN ERROR MESSAGE.
1898  4508                          SP
 (1)                                ;*****************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1899                                ;*BGNSUB
1900                                ;*   IF CHANNEL 0=ILLEGAL 5X4
1901                                ;*     THEN-CONTINUE
1902                                ;*     ELSE-ERROR CHANNEL 0
1903                                ;*   ENDIF
1904                                ;*   IF CHANNEL 1=ILLEGAL 5X4
1905                                ;*     THEN-CONTINUE
1906                                ;*     ELSE-ERROR CHANNEL 1
1907                                ;*   ENDIF
1908                                ;*   IF CHANNEL 2=ILLEGAL 5X4
1909                                ;*     THEN-CONTINUE
1910                                ;*     ELSE-ERROR CHANNEL 2
1911                                ;*   ENDIF
1912                                ;*   IF CHANNEL 3-ILLEGAL 5X4
1913                                ;*     THEN-CONTINUE
1914                                ;*     ELSE-ERROR CHANNEL 3
1915                                ;*   ENDIF
1916                                ;*   IF CHANNEL 4-ILLEGAL 5X4
1917                                ;*     THEN-CONTINUE
1918                                ;*     ELSE-ERROR CHANNEL 4
1919                                ;*   ENDIF
1920                                ;*   IF CHANNEL 5=ILLEGAL 5X4
1921                                ;*     THEN-CONTINUE
1922                                ;*     ELSE-ERROR CHANNEL 5
1923                                ;*   ENDIF
1924                                ;*   IF CHANNEL 6-ILLEGAL 5X4
1925                                ;*     THEN-CONTINUE
1926                                ;*     ELSE-ERROR CHANNEL 6
1927                                ;*   ENDIF
1928                                ;*   IF CHANNEL 7=ILLEGAL 5X4
1929                                ;*     THEN-CONTINUE
1930                                ;*     ELSE-ERROR CHANNEL 7
1931                                ;*   ENDIF
1932                                ;*   IF CHANNEL P=ILLEGAL 5X4
1933                                ;*     THEN-CONTINUE
1934                                ;*     ELSE-ERROR CHANNEL P
1935                                ;*   ENDIF
1936                                ;*ENDSUB
```

```
1937  450B                                  S
 (1)                                        ; ***************************************************************************
1938
1939  450B  DB  12          10.0  CKILL:  IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1940  450D  E6  01           7.0          ANI     $01             ;CHANNEL 0 ILLEGAL?
1941  450F  C2  17  45       10.0         JNZ     CKIL0           ;YES-CONTINUE
1942  4512                               ERR     OUTILL,CKIL0
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4512  CD  09  28        8.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0003                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4515  03                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4516  00                              .BYTE
 (1)  4517  CD  15  28        18.0                CKIL0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  451A  DA  AD  45        10.0                        JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1943                                    ;>CHANNEL 0 NOT ILLEGAL 5X4
1944
1945  451D  DB  12          10.0         IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1946  451F  E6  02           7.0         ANI     $02             ;CHANNEL 1 ILLEGAL?
1947  4521  C2  29  45       10.0         JNZ     CKIL1           ;YES-CONTINUE
1948  4524                               ERR     OUTILL,CKIL1
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4524  CD  09  28        18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                            MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4527  04                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4528  0C                              .BYTE
 (1)  4529  CD  15  28        18.0                CKIL1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  452C  DA  AD  45        10.0                        JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1949                                    ;>CHANNEL 1 NOT ILLEGAL 5X4
1950
1951  452F  DB  12          10.0         IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1952  4531  E6  04           7.0         ANI     $04             ;CHANNEL 2 ILLEGAL?
1953  4533  C2  3B  45       10.0         JNZ     CKIL2           ;YES-CONTINUE
1954  4536                               ERR     OUTILL,CKIL2
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4536  CD  09  28        18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4539  05                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453A  00                              .BYTE
 (1)  453B  CD  15  28        18.0                CKIL2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  453E  DA  AD  45        10.0                        JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1955                                    ;>CHANNEL 2 NOT ILLEGAL 5X4
1956
1957  4541  DB  12          10.0         IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1958  4543  E6  08           7.0         ANI     $08             ;CHANNEL 3 ILLEGAL?
1959  4545  C2  4D  45       10.0         JNZ     CKIL3           ;YES-CONTINUE
1960  4548                               ERR     OUTILL,CKIL3
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4548  CD  09  28        18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454B  06                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454C  00                              .BYTE
 (1)  454D  CD  15  28        18.0                CKIL3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4550  DA  AD  45        10.0                        JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1961                                    ;>CHANNEL 3 NOT ILLEGAL 5X4
```

```
1962  4553  DB  12          10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1963  4555  E6  10           7.0        ANI     $10             ;CHANNEL 4 ILLEGAL?
1964  4557  C2  5F  45      10.0        JNZ     CKIL4           ;YES-CONTINUE
1965  455A                              ERR     OUTILL,CKIL4
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  455A  CD  09  28      18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  455D  07                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  455E  00                              .BYTE
 (1)  455F  CD  15  28      18.0        CKIL4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4562  DA  AD  45      10.0                JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1966                        ;>CHANNEL 4 NOT ILLEGAL 5X4
1967
1968  4565  DB  12          10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1969  4567  E6  20           7.0        ANI     $20             ;CHANNEL 5 ILLEGAL?
1970  4569  C2  71  45      10.0        JNZ     CKIL5           ;YES-CONTINUE
1971  456C                              ERR     OUTILL,CKIL5
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  456C  CD  09  28      18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  456F  08                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4570  00                              .BYTE
 (1)  4571  CD  15  28      18.0        CKIL5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4574  DA  AD  45      10.0                JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1972                        ;>CHANNEL 5 NOT ILLEGAL 5X4
1973  4577  DB  12          10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1974  4579  E6  40           7.0        ANI     $40             ;CHANNEL 6 ILLEGAL?
1975  457B  C2  83  45      10.0        JNZ     CKIL6           ;YES-CONTINUE
1976  457E                              ERR     OUTILL,CKIL6
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  457E  CD  09  28      18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0009                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4581  09                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4582  00                              .BYTE
 (1)  4583  CD  15  28      18.0        CKIL6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4586  DA  AD  45      10.0                JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1977                        ;>CHANNEL 6 NOT ILLEGAL 5X4
1978
1979  4589  DB  12          10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1980  458B  E6  80           7.0        ANI     $80             ;CHANNEL 7 ILLEGAL?
1981  458D  C2  95  45      10.0        JNZ     CKIL7           ;YES-CONTINUE
1982  4590                              ERR     OUTILL,CKIL7
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4590  CD  09  28      18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4593  0A                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4594  00                              .BYTE
 (1)  4595  CD  15  28      18.0        CKIL7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4598  DA  AD  45      10.0                JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1983                        ;>CHANNEL 7 NOT ILLEGAL 5X4
```

```
1985  459B  DB  15        10.0          IN      RPSTA           ;GET THE CHANNEL P BIT
1986  459D  E6  04         7.0          ANI     R.ILL           ;CHANNEL P ILLEGAL?
1987  459F  C2  A7  45     10.0          JNZ     CKILP           ;YES-CONTINUE
1988  45A2                               ERR     OUTILL,CKILP
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45A2  CD  09  28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45A5  08                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45A6  00                                  .BYTE
 (1)  45A7  CD  15  28     18.0          CKILP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45AA  DA  AD  45     10.0                  JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1989                                ;>CHANNEL P NOT ILLEGAL 5X4
1990
1991  45AD  C9                10.0  OUTILL: RET                     ;RETURN
```

```
                                .SBTTL  SUBROUTINE CHECK 5X4
45AE                            SSUB
                                ;*****************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK 5X4 SUBROUTINE
45AE                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ CHANNEL DATA
                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
45AE                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*      IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
                                ;*        THEN-ERROR CHANNEL 0
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
                                ;*        THEN-ERROR CHANNEL 1
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
                                ;*        THEN-ERROR CHANNEL 2
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
                                ;*        THEN-ERROR CHANNEL 3
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL 4 ACTUAL INFORMATION NOT - CHANNEL 4 EXPECTED
                                ;*        THEN-ERROR CHANNEL 4
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
                                ;*        THEN-ERROR CHANNEL 5
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
                                ;*        THEN-ERROR CHANNEL 6
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL 7 ACTUAL INFORMATION NOT - CHANNEL 7 EXPECTED
                                ;*        THEN-ERROR CHANNEL 7
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      IF CHANNEL P ACTUAL INFORMATION NOT = CHANNEL P EXPECTED
                                ;*        THEN-ERROR CHANNEL P
                                ;*        ELSE-CONTINUE
                                ;*      ENDIF
```

```
2038                                        ;*ENDSUB
2039  45AE                                  S
 (1)                                        ; ************************************************************************************
2040  45AE  21  4B  48      10.0  CK5X4:  LXI   H,DATAE          ;LOAD EXPECTED DATA ADDRESS
2041  45B1  3A  49  48      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
2042  45B4                               ROUT  ADATA            ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  45B4  D3  94            10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  45B6  7F                 4.0                MOV   A,A            ;RETRY LINK
2043  45B7  E6  01            7.0           ANI   $01              ;REMOVE UNWANTED BITS
2044  45B9  47                 4.0           MOV   B,A              ;SAVE IN B
2045  45BA  7E                 7.0           MOV   A,M              ;GET EXPECTED DATA
2046  45BB                               ROUT  EDATA            ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  45BB  D3  95            10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  45BD  7F                 4.0                MOV   A,A            ;RETRY LINK
2047  45BE  E6  01            7.0           ANI   $01              ;REMOVE UNWANTED BITS
2048  45C0  B8                 4.0           CMP   B                ;COMPARE
2049  45C1  CA  C9  45        10.0           JZ    CKAM0            ;CONTINUE IF EQUAL
2050  45C4                               ERRB  OUT,CKAM0,@7
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  45C4  CD  12  28        18.0                CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)      000C                                    MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45C7  0C                                    .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  45C8  07                                    .BYTE @7             ;PRINT ROUTINE NUMBER
 (1)  45C9  CD  15  28        18.0          CKAM0:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45CC  DA  92  46        10.0                JC    OUT            ;LOOP ADDRESS IF LOOP SPECIFIED
2051                                       ;>CHANNEL 0 FAILED
2052
2053  45CF  3A  49  48      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
2054  45D2  E6  02            7.0           ANI   $02              ;REMOVE UNWANTED BITS
2055  45D4  47                 4.0           MOV   B,A              ;SAVE IN B
2056  45D5  7E                 7.0           MOV   A,M              ;GET EXPECTED DATA
2057  45D6  E6  02            7.0           ANI   $02              ;REMOVE UNWANTED BITS
2058  45D8  B8                 4.0           CMP   B                ;COMPARE
2059  45D9  CA  E1  45        10.0           JZ    CKAM1            ;CONTINUE IF EQUAL
2060  45DC                               ERRB  OUT,CKAM1,@7
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  45DC  CD  12  28        18.0                CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)      000D                                    MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45DF  0D                                    .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  45E0  07                                    .BYTE @7             ;PRINT ROUTINE NUMBER
 (1)  45E1  CD  15  28        18.0          CKAM1:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45E4  DA  92  46        10.0                JC    OUT            ;LOOP ADDRESS IF LOOP SPECIFIED
2061                                       ;>CHANNEL 1 FAILED
```

```
2063  45E7  3A 49 48        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2064  45EA  E6 04             7.0          ANI     $04             ;REMOVE UNWANTED BITS
2065  45EC  47                 4.0          MOV     B,A             ;SAVE IN B
2066  45ED  7E                 7.0          MOV     A,M             ;GET EXPECTED DATA
2067  45EE  E6 04              7.0          ANI     $04             ;REMOVE UNWANTED BITS
2068  45F0  B8                 4.0          CMP     B               ;COMPARE
2069  45F1  CA F9 45          10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
2070  45F4                                  ERRB    OUT,CKAM2,@7
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  45F4  CD 12 28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000E                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45F7  0E                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45F8  07                                      .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  45F9  CD 15 28          18.0          CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45FC  DA 92 46          10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2071                                  ;>CHANNEL 2 FAILED
2072
2073  45FF  3A 49 48        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2074  4602  E6 08             7.0          ANI     $08             ;REMOVE UNWANTED BITS
2075  4604  47                 4.0          MOV     B,A             ;SAVE IN B
2076  4605  7E                 7.0          MOV     A,M             ;GET EXPECTED DATA
2077  4606  E6 08              7.0          ANI     $08             ;REMOVE UNWANTED BITS
2078  4608  B8                 4.0          CMP     B               ;COMPARE
2079  4609  CA 11 46          10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2080  460C                                  ERRB    OUT,CKAM3,@7
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  460C  CD 12 28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000F                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460F  0F                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4610  07                                      .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4611  CD 15 28          18.0          CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4614  DA 92 46          10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2081                                  ;>CHANNEL 3 FAILED
2082
2083  4617  3A 49 48        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2084  461A  E6 10              7.0          ANI     $10             ;REMOVE UNWANTED BITS
2085  461C  47                 4.0          MOV     B,A             ;SAVE IN B
2086  461D  7E                 7.0          MOV     A,M             ;GET EXPECTED DATA
2087  461E  E6 10              7.0          ANI     $10             ;REMOVE UNWANTED BITS
2088  4620  B8                 4.0          CMP     B               ;COMPARE
2089  4621  CA 29 46          10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2090  4624                                  ERRB    OUT,CKAM4,@7
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4624  CD 12 28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0010                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4627  10                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4628  07                                      .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4629  CD 15 28          18.0          CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  462C  DA 92 46          10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2091                                  ;>CHANNEL 4 FAILED
```

```
2093  462F  3A 49 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2094  4632  E6 20            7.0          ANI     $20             ;REMOVE UNWANTED BITS
2095  4634  47               4.0          MOV     B,A             ;SAVE IN B
2096  4635  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
2097  4636  E6 20            7.0          ANI     $20             ;REMOVE UNWANTED BITS
2098  4638  B8               4.0          CMP     B               ;COMPARE
2099  4639  CA 41 46        10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
2100  463C                                ERRB    OUT,CKAM5,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  463C  CD 12 28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0011                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  463F  11                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4640  07                               .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4641  CD 15 28      18.0          CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4644  DA 92 46      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2101                                ;>CHANNEL 5 FAILED
2102
2103  4647  3A 49 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2104  464A  E6 40            7.0          ANI     $40             ;REMOVE UNWANTED BITS
2105  464C  47               4.0          MOV     B,A             ;SAVE IN B
2106  464D  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
2107  464E  E6 40            7.0          ANI     $40             ;REMOVE UNWANTED BITS
2108  4650  B8               4.0          CMP     B               ;COMPARE
2109  4651  CA 59 46        10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
2110  4654                                ERRB    OUT,CKAM6,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4654  CD 12 28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0012                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4657  12                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4658  07                               .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4659  CD 15 28      18.0          CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  465C  DA 92 46      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2111                                ;>CHANNEL 6 FAILED
2112
2113  465F  3A 49 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2114  4662  E6 80            7.0          ANI     $80             ;REMOVE UNWANTED BITS
2115  4664  47               4.0          MOV     B,A             ;SAVE IN B
2116  4665  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
2117  4666  E6 80            7.0          ANI     $80             ;REMOVE UNWANTED BITS
2118  4668  B8               4.0          CMP     B               ;COMPARE
2119  4669  CA 71 46        10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
2120  466C                                ERRB    OUT,CKAM7,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  466C  CD 12 28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0013                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  466F  13                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4670  07                               .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4671  CD 15 28      18.0          CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4674  DA 92 46      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2121                                ;>CHANNEL 7 FAILED
```

```
 2123  4677  21  4C  48      10.0          LXI     H,DATAEP        ;GET THE EXPECTED DATA PARITY ADDRESS
 2124  467A  3A  4A  48      13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
 2125  467D  E6  01           7.0          ANI     $01             ;REMOVE UNWANTED BITS
 2126  467F  47               4.0          MOV     B,A             ;SAVE IN B
 2127  4680  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
 2128  4681  E6  01           7.0          ANI     $01             ;REMOVE UNWANTED BITS
 2129  4683  B8               4.0          CMP     B               ;COMPARE
 2130  4684  CA  8C  46      10.0          JZ      CKAMP           ;CONTINUE IF EQUAL
 2131  4687                                ERRB    OUT,CKAMP,@7
   (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
   (1)  4687  CD  12  28      18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
   (1)        0014                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
   (1)  468A  14                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
   (1)  468B  07                                       .BYTE   @7              ;PRINT ROUTINE NUMBER
   (1)  468C  CD  15  28      18.0                CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
   (1)  468F  DA  92  46      10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
 2132                                  ;>CHANNEL P FAILED
 2133
 2134  4692  C9               10.0  OUT:     RET
```

```
                                .SBTTL  TABLE INPUT DATA

M8950E: .BYTE   @000            ;DATA GROUP 1 - CHARACTER 1
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 1  PARITY
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 2
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 2  PARITY
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3  PARITY
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4  PARITY
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 5
        .BYTE   @000            ;DATA GROUP 1 - CHARACTER 5  PARITY

        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 1
        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 1  PARITY
        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 2
        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 2  PARITY
        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3
        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3  PARITY
        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 4
        .BYTE   @000            ;DATA GROUP 2 - CHARACTER 4  PARITY
        .BYTE   @377            ;DATA GROUP 2 - CHARACTER 5
        .BYTE   @001            ;DATA GROUP 2 - CHARACTER 5  PARITY

        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 1
        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 1  PARITY
        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2
        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2  PARITY
        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3
        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3  PARITY
        .BYTE   @377            ;DATA GROUP 3 - CHARACTER 4
        .BYTE   @001            ;DATA GROUP 3 - CHARACTER 4  PARITY
        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5
        .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5  PARITY

        .BYTE   @000            ;DATA GROUP 4 - CHARACTER 1
        .BYTE   @000            ;DATA GROUP 4 - CHARACTER 1  PARITY
        .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2
        .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2  PARITY
        .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3
        .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3  PARITY
        .BYTE   @377            ;DATA GROUP 4 - CHARACTER 4
        .BYTE   @001            ;DATA GROUP 4 - CHARACTER 4  PARITY
        .BYTE   @377            ;DATA GROUP 4 - CHARACTER 5
        .BYTE   @001            ;DATA GROUP 4 - CHARACTER 5  PARITY

        .BYTE   @000            ;DATA GROUP 5 - CHARACTER 1
        .BYTE   @000            ;DATA GROUP 5 - CHARACTER 1  PARITY
        .BYTE   @000            ;DATA GROUP 5 - CHARACTER 2
        .BYTE   @000            ;DATA GROUP 5 - CHARACTER 2  PARITY
        .BYTE   @377            ;DATA GROUP 5 - CHARACTER 3
        .BYTE   @001            ;DATA GROUP 5 - CHARACTER 3  PARITY
        .BYTE   @000            ;DATA GROUP 5 - CHARACTER 4
        .BYTE   @000            ;DATA GROUP 5 - CHARACTER 4  PARITY
```

```
2136
2137
2138  4693  00
2139  4694  00
2140  4695  00
2141  4696  00
2142  4697  00
2143  4698  00
2144  4699  00
2145  469A  00
2146  469B  00
2147  469C  00
2148
2149  469D  00
2150  469E  00
2151  469F  00
2152  46A0  00
2153  46A1  00
2154  46A2  00
2155  46A3  00
2156  46A4  00
2157  46A5  FF
2158  46A6  01
2159
2160  46A7  00
2161  46A8  00
2162  46A9  00
2163  46AA  00
2164  46AB  00
2165  46AC  00
2166  46AD  FF
2167  46AE  01
2168  46AF  00
2169  46B0  00
2170
2171  46B1  00
2172  46B2  00
2173  46B3  00
2174  46B4  00
2175  46B5  00
2176  46B6  00
2177  46B7  FF
2178  46B8  01
2179  46B9  FF
2180  46BA  01
2181
2182  46BB  00
2183  46BC  00
2184  46BD  00
2185  46BE  00
2186  46BF  FF
2187  46C0  01
2188  46C1  00
2189  46C2  00
```

```
2190  46C3  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 5
2191  46C4  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 5  PARITY
2192
2193  46C5  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 1
2194  46C6  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 1  PARITY
2195  46C7  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2
2196  46C8  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2  PARITY
2197  46C9  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 3
2198  46CA  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 3  PARITY
2199  46CB  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4
2200  46CC  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4  PARITY
2201  46CD  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 5
2202  46CE  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 5  PARITY
2203
2204  46CF  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 1
2205  46D0  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 1  PARITY
2206  46D1  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2
2207  46D2  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2  PARITY
2208  46D3  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 3
2209  46D4  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 3  PARITY
2210  46D5  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 4
2211  46D6  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 4  PARITY
2212  46D7  C0          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5
2213  46D8  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5  PARITY
2214
2215  46D9  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 1
2216  46DA  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 1  PARITY
2217  46DB  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2
2218  46DC  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2  PARITY
2219  46DD  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 3
2220  46DE  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 3  PARITY
2221  46DF  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 4
2222  46E0  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 4  PARITY
2223  46E1  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 5
2224  46E2  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 5  PARITY
2225
2226  46E3  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 1
2227  46E4  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 1  PARITY
2228  46E5  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 2
2229  46E6  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 2  PARITY
2230  46E7  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3
2231  46E8  00          .BYTE   @000          :DATA GROUP 9 - CHARACTER 3  PARITY
2232  46E9  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 4
2233  46EA  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 4  PARITY
2234  46EB  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5
2235  46EC  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5  PARITY
2236
2237  46ED  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1
2238  46EE  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1  PARITY
2239  46EF  FF          .BYTE   @377          ;DATA GROUP 10 - CHARACTER 2
2240  46F0  01          .BYTE   @001          ;DATA GROUP 10 - CHARACTER 2  PARITY
2241  46F1  FF          .BYTE   @377          ;DATA GROUP 10 - CHARACTER 3
2242  46F2  01          .BYTE   @001          ;DATA GROUP 10 - CHARACTER 3  PARITY
2243  46F3  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 4
```

```
2244  46F4  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 4  PARITY
2245  46F5  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 5
2246  46F6  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 5  PARITY
2247
2248  46F7  FF          .BYTE   @377           ;DATA GROUP 11 - CHARACTER 1
2249  46F8  01          .BYTE   @001           ;DATA GROUP 11 - CHARACTER 1  PARITY
2250  46F9  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 2
2251  46FA  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 2  PARITY
2252  46FB  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3
2253  46FC  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3  PARITY
2254  46FD  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 4
2255  46FE  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 4  PARITY
2256  46FF  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5
2257  4700  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5  PARITY
2258
2259  4701  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 1
2260  4702  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 1  PARITY
2261  4703  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 2
2262  4704  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 2  PARITY
2263  4705  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3
2264  4706  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3  PARITY
2265  4707  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 4
2266  4708  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 4  PARITY
2267  4709  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 5
2268  470A  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 5  PARITY
2269
2270  470B  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 1
2271  470C  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 1  PARITY
2272  470D  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 2
2273  470E  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 2  PARITY
2274  470F  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 3
2275  4710  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 3  PARITY
2276  4711  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 4
2277  4712  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 4  PARITY
2278  4713  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5
2279  4714  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5  PARITY
2280
2281  4715  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 1
2282  4716  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 1  PARITY
2283  4717  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 2
2284  4718  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 2  PARITY
2285  4719  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 3
2286  471A  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 3  PARITY
2287  471B  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4
2288  471C  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4  PARITY
2289  471D  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 5
2290  471E  80          .BYTE   @200           ;DATA GROUP 14 - CHARACTER 5  PARITY
2291
```

```
                                        .SBTTL  TABLE UNTRANSLATED INPUT DATA

                                ;INPUT DATA TABLE - THIS DATA IS LOADED INTO THE READ CHANNEL FIFO'S
                                ;ONE GROUP AT A TIME AND THEN TRANSLATED.

2298  471F  FF         M8950I: .BYTE   @377          ;DATA GROUP 1 - CHARACTER 1
2299  4720  01                 .BYTE   @001          ;DATA GROUP 1 - CHARACTER 1  PARITY
2300  4721  FF                 .BYTE   @377          ;DATA GROUP 1 - CHARACTER 2
2301  4722  01                 .BYTE   @001          ;DATA GROUP 1 - CHARACTER 2  PARITY
2302  4723  00                 .BYTE   @000          ;DATA GROUP 1 - CHARACTER 3
2303  4724  00                 .BYTE   @000          ;DATA GROUP 1 - CHARACTER 3  PARITY
2304  4725  00                 .BYTE   @000          ;DATA GROUP 1 - CHARACTER 4
2305  4726  00                 .BYTE   @000          ;DATA GROUP 1 - CHARACTER 4  PARITY
2306  4727  FF                 .BYTE   @377          ;DATA GROUP 1 - CHARACTER 5
2307  4728  01                 .BYTE   @001          ;DATA GROUP 1 - CHARACTER 5  PARITY
2308
2309  4729  FF                 .BYTE   @377          ;DATA GROUP 2 - CHARACTER 1
2310  472A  01                 .BYTE   @001          ;DATA GROUP 2 - CHARACTER 1  PARITY
2311  472B  FF                 .BYTE   @377          ;DATA GROUP 2 - CHARACTER 2
2312  472C  01                 .BYTE   @001          ;DATA GROUP 2 - CHARACTER 2  PARITY
2313  472D  00                 .BYTE   @000          ;DATA GROUP 2 - CHARACTER 3
2314  472E  00                 .BYTE   @000          ;DATA GROUP 2 - CHARACTER 3  PARITY
2315  472F  FF                 .BYTE   @377          ;DATA GROUP 2 - CHARACTER 4
2316  4730  01                 .BYTE   @001          ;DATA GROUP 2 - CHARACTER 4  PARITY
2317  4731  FF                 .BYTE   @377          ;DATA GROUP 2 - CHARACTER 5
2318  4732  01                 .BYTE   @001          ;DATA GROUP 2 - CHARACTER 5  PARITY
2319
2320  4733  FF                 .BYTE   @377          ;DATA GROUP 3 - CHARACTER 1
2321  4734  01                 .BYTE   @001          ;DATA GROUP 3 - CHARACTER 1  PARITY
2322  4735  00                 .BYTE   @000          ;DATA GROUP 3 - CHARACTER 2
2323  4736  00                 .BYTE   @000          ;DATA GROUP 3 - CHARACTER 2  PARITY
2324  4737  00                 .BYTE   @000          ;DATA GROUP 3 - CHARACTER 3
2325  4738  00                 .BYTE   @000          ;DATA GROUP 3 - CHARACTER 3  PARITY
2326  4739  FF                 .BYTE   @377          ;DATA GROUP 3 - CHARACTER 4
2327  473A  01                 .BYTE   @001          ;DATA GROUP 3 - CHARACTER 4  PARITY
2328  473B  00                 .BYTE   @000          ;DATA GROUP 3 - CHARACTER 5
2329  473C  00                 .BYTE   @000          ;DATA GROUP 3 - CHARACTER 5  PARITY
2330
2331  473D  FF                 .BYTE   @377          ;DATA GROUP 4 - CHARACTER 1
2332  473E  01                 .BYTE   @001          ;DATA GROUP 4 - CHARACTER 1  PARITY
2333  473F  00                 .BYTE   @000          ;DATA GROUP 4 - CHARACTER 2
2334  4740  00                 .BYTE   @000          ;DATA GROUP 4 - CHARACTER 2  PARITY
2335  4741  00                 .BYTE   @000          ;DATA GROUP 4 - CHARACTER 3
2336  4742  00                 .BYTE   @000          ;DATA GROUP 4 - CHARACTER 3  PARITY
2337  4743  FF                 .BYTE   @377          ;DATA GROUP 4 - CHARACTER 4
2338  4744  01                 .BYTE   @001          ;DATA GROUP 4 - CHARACTER 4  PARITY
2339  4745  FF                 .BYTE   @377          ;DATA GROUP 4 - CHARACTER 5
2340  4746  01                 .BYTE   @001          ;DATA GROUP 4 - CHARACTER 5  PARITY
2341
2342  4747  FF                 .BYTE   @377          ;DATA GROUP 5 - CHARACTER 1
2343  4748  01                 .BYTE   @001          ;DATA GROUP 5 - CHARACTER 1  PARITY
2344  4749  FF                 .BYTE   @377          ;DATA GROUP 5 - CHARACTER 2
2345  474A  01                 .BYTE   @001          ;DATA GROUP 5 - CHARACTER 2  PARITY
2346  474B  FF                 .BYTE   @377          ;DATA GROUP 5 - CHARACTER 3
```

```
2347  474C  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 3  PARITY
2348  474D  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4
2349  474E  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4  PARITY
2350  474F  FF          .BYTE   @377          ;DATA GROUP 5 - CHARACTER 5
2351  4750  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 5  PARITY
2352
2353  4751  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 1
2354  4752  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 1  PARITY
2355  4753  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2
2356  4754  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2  PARITY
2357  4755  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 3
2358  4756  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 3  PARITY
2359  4757  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4
2360  4758  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4  PARITY
2361  4759  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 5
2362  475A  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 5  PARITY
2363
2364  475B  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 1
2365  475C  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 1  PARITY
2366  475D  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2
2367  475E  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2  PARITY
2368  475F  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 3
2369  4760  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 3  PARITY
2370  4761  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 4
2371  4762  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 4  PARITY
2372  4763  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5
2373  4764  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5  PARITY
2374
2375  4765  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 1
2376  4766  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 1  PARITY
2377  4767  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2
2378  4768  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2  PARITY
2379  4769  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 3
2380  476A  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 3  PARITY
2381  476B  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 4
2382  476C  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 4  PARITY
2383  476D  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 5
2384  476E  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 5  PARITY
2385
2386  476F  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 1
2387  4770  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 1  PARITY
2388  4771  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 2
2389  4772  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 2  PARITY
2390  4773  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3
2391  4774  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3  PARITY
2392  4775  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 4
2393  4776  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 4  PARITY
2394  4777  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5
2395  4778  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5  PARITY
2396
2397  4779  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1
2398  477A  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1  PARITY
2399  477B  FF          .BYTE   @377          ;DATA GROUP 10 - CHARACTER 2
2400  477C  01          .BYTE   @001          ;DATA GROUP 10 - CHARACTER 2  PARITY
```

```
2401  477D  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 3
2402  477E  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 3  PARITY
2403  477F  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 4
2404  4780  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 4  PARITY
2405  4781  FF          .BYTE   @377          ;DATA GROUP 10 - CHARACTER 5
2406  4782  01          .BYTE   @001          ;DATA GROUP 10 - CHARACTER 5  PARITY
2407
2408  4783  00          .BYTE   @000          ;DATA GROUP 11 - CHARACTER 1
2409  4784  00          .BYTE   @000          ;DATA GROUP 11 - CHARACTER 1  PARITY
2410  4785  FF          .BYTE   @377          ;DATA GROUP 11 - CHARACTER 2
2411  4786  01          .BYTE   @001          ;DATA GROUP 11 - CHARACTER 2  PARITY
2412  4787  00          .BYTE   @000          ;DATA GROUP 11 - CHARACTER 3
2413  4788  00          .BYTE   @000          ;DATA GROUP 11 - CHARACTER 3  PARITY
2414  4789  FF          .BYTE   @377          ;DATA GROUP 11 - CHARACTER 4
2415  478A  01          .BYTE   @001          ;DATA GROUP 11 - CHARACTER 4  PARITY
2416  478B  00          .BYTE   @000          ;DATA GROUP 11 - CHARACTER 5
2417  478C  00          .BYTE   @000          ;DATA GROUP 11 - CHARACTER 5  PARITY
2418
2419  478D  00          .BYTE   @000          ;DATA GROUP 12 - CHARACTER 1
2420  478E  00          .BYTE   @000          ;DATA GROUP 12 - CHARACTER 1  PARITY
2421  478F  FF          .BYTE   @377          ;DATA GROUP 12 - CHARACTER 2
2422  4790  01          .BYTE   @001          ;DATA GROUP 12 - CHARACTER 2  PARITY
2423  4791  C0          .BYTE   @000          ;DATA GROUP 12 - CHARACTER 3
2424  4792  00          .BYTE   @000          ;DATA GROUP 12 - CHARACTER 3  PARITY
2425  4793  FF          .BYTE   @377          ;DATA GROUP 12 - CHARACTER 4
2426  4794  01          .BYTE   @001          ;DATA GROUP 12 - CHARACTER 4  PARITY
2427  4795  FF          .BYTE   @377          ;DATA GROUP 12 - CHARACTER 5
2428  4796  01          .BYTE   @001          ;DATA GROUP 12 - CHARACTER 5  PARITY
2429
2430  4797  FF          .BYTE   @377          ;DATA GROUP 13 - CHARACTER 1
2431  4798  01          .BYTE   @001          ;DATA GROUP 13 - CHARACTER 1  PARITY
2432  4799  FF          .BYTE   @377          ;DATA GROUP 13 - CHARACTER 2
2433  479A  01          .BYTE   @001          ;DATA GROUP 13 - CHARACTER 2  PARITY
2434  479B  FF          .BYTE   @377          ;DATA GROUP 13 - CHARACTER 3
2435  479C  01          .BYTE   @001          ;DATA GROUP 13 - CHARACTFR 3  PARITY
2436  479D  FF          .BYTE   @377          ;DATA GROUP 13 - CHARACTER 4
2437  479E  01          .BYTE   @001          ;DATA GROUP 13 - CHARACTER 4  PARITY
2438  479F  00          .BYTE   @000          ;DATA GROUP 13 - CHARACTER 5
2439  47A0  00          .BYTE   @000          ;DATA GROUP 13 - CHARACTER 5  PARITY
2440
2441  47A1  00          .BYTE   @000          ;DATA GROUP 14 - CHARACTER 1
2442  47A2  00          .BYTE   @000          ;DATA GROUP 14 - CHARACTER 1  PARITY
2443  47A3  FF          .BYTE   @377          ;DATA GROUP 14 - CHARACTER 2
2444  47A4  01          .BYTE   @001          ;DATA GROUP 14 - CHARACTER 2  PARITY
2445  47A5  FF          .BYTE   @377          ;DATA GROUP 14 - CHARACTER 3
2446  47A6  01          .BYTE   @001          ;DATA GROUP 14 - CHARACTER 3  PARITY
2447  47A7  00          .BYTE   @000          ;DATA GROUP 14 - CHARACTER 4
2448  47A8  00          .BYTE   @000          ;DATA GROUP 14 - CHARACTER 4  PARITY
2449  47A9  FF          .BYTE   @377          ;DATA GROUP 14 - CHARACTER 5
2450  47AA  01          .BYTE   @001          ;DATA GROUP 14 - CHARACTER 5  PARITY
2451
2452  47AB  00          .BYTE   @000          ;DATA GROUP 15 - CHARACTER 1
2453  47AC  00          .BYTE   @000          ;DATA GROUP 15 - CHARACTER 1  PARITY
2454  47AD  FF          .BYTE   @377          ;DATA GROUP 15 - CHARACTER 2
```

```
2455  47AE  01          .BYTE   @001        ;DATA GROUP 15 - CHARACTER 2  PARITY
2456  47AF  FF          .BYTE   @377        ;DATA GROUP 15 - CHARACTER 3
2457  47B0  01          .BYTE   @001        ;DATA GROUP 15 - CHARACTER 3  PARITY
2458  47B1  FF          .BYTE   @377        ;DATA GROUP 15 - CHARACTER 4
2459  47B2  01          .BYTE   @001        ;DATA GROUP 15 - CHARACTER 4  PARITY
2460  47B3  00          .BYTE   @000        ;DATA GROUP 15 - CHARACTER 5
2461  47B4  00          .BYTE   @000        ;DATA GROUP 15 - CHARACTER 5  PARITY
2462
2463  47B5  00          .BYTE   @000        ;DATA GROUP 16 - CHARACTER 1
2464  47B6  00          .BYTE   @000        ;DATA GROUP 16 - CHARACTER 1  PARITY
2465  47B7  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 2
2466  47B8  01          .BYTE   @001        ;DATA GROUP 16 - CHARACTER 2  PARITY
2467  47B9  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 3
2468  47BA  01          .BYTE   @001        ;DATA GROUP 16 - CHARACTER 3  PARITY
2469  47BB  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 4
2470  47BC  01          .BYTE   @001        ;DATA GROUP 16 - CHARACTER 4  PARITY
2471  47BD  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 5
2472  47BE  81          .BYTE   @201        ;DATA GROUP 16 - CHARACTER 5  PARITY
2473
```

```
                                .SBTTL  TABLE TRANSLATED OUTPUT DATA REVERSE
                                ;OUTPUT DATA TABLE - THIS DATA IS COMPARED WITH THE OUTPUT OF THE READ
                                ;CHANNEL BOARDS WHEN DATA READY SETS.  IT REFLECTS THE TRANSLATED INPUT
                                ;DATA ON SUBGROUP BASIS.

47BF  FF        M89500: .BYTE   @377            ;DATA GROUP 1 - DATA VALUE 1
47C0  01                .BYTE   @001            ;DATA GROUP 1 - DATA VALUE 1 PARITY
47C1  FF                .BYTE   @377            ;DATA GROUP 1 - DATA VALUE 2
47C2  01                .BYTE   @001            ;DATA GROUP 1 - DATA VALUE 2 PARITY
47C3  00                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3
47C4  00                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3 PARITY
47C5  00                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4
47C6  00                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4 PARITY

47C7  FF                .BYTE   @377            ;DATA GROUP 2 - DATA VALUE 1
47C8  01                .BYTE   @001            ;DATA GROUP 2 - DATA VALUE 1 PARITY
47C9  00                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2
47CA  00                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2 PARITY
47CB  00                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3
47CC  00                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3 PARITY
47CD  00                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 4
47CE  00                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 4 PARITY

47CF  FF                .BYTE   @377            ;DATA GROUP 3 - DATA VALUE 1
47D0  01                .BYTE   @001            ;DATA GROUP 3 - DATA VALUE 1 PARITY
47D1  00                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2
47D2  00                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2 PARITY
47D3  00                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 3
47D4  00                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 3 PARITY
47D5  FF                .BYTE   @377            ;DATA GROUP 3 - DATA VALUE 4
47D6  01                .BYTE   @C01            ;DATA GROUP 3 - DATA VALUE 4 PARITY

47D7  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1
47D8  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1 PARITY
47D9  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2
47DA  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2 PARITY
47DB  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 3
47DC  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 3 PARITY
47DD  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 4
47DE  00                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 4 PARITY

47DF  FF                .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 1
47E0  01                .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 1 PARITY
47E1  FF                .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 2
47E2  01                .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 2 PARITY
47E3  FF                .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 3
47E4  01                .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 3 PARITY
47E5  00                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4
47E6  00                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4 PARITY

47E7  FF                .BYTE   @377            ;DATA GROUP 6 - DATA VALUE 1
47E8  01                .BYTE   @001            ;DATA GROUP 6 - DATA VALUE 1 PARITY
47E9  00                .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 2
47EA  00                .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 2 PARITY
```

```
2529  47EB  FF      .BYTE  @377   ;DATA GROUP 6 - DATA VALUE 3
2530  47EC  01      .BYTE  @001   ;DATA GROUP 6 - DATA VALUE 3 PARITY
2531  47ED  00      .BYTE  @000   ;DATA GROUP 6 - DATA VALUE 4
2532  47EE  00      .BYTE  @000   ;DATA GROUP 6 - DATA VALUE 4 PARITY
2533
2534  47EF  FF      .BYTE  @377   ;DATA GROUP 7 - DATA VALUE 1
2535  47F0  01      .BYTE  @001   ;DATA GROUP 7 - DATA VALUE 1 PARITY
2536  47F1  00      .BYTE  @000   ;DATA GROUP 7 - DATA VALUE 2
2537  47F2  00      .BYTE  @000   ;DATA GROUP 7 - DATA VALUE 2 PARITY
2538  47F3  FF      .BYTE  @377   ;DATA GROUP 7 - DATA VALUE 3
2539  47F4  01      .BYTE  @001   ;DATA GROUP 7 - DATA VALUE 3 PARITY
2540  47F5  FF      .BYTE  @377   ;DATA GROUP 7 - DATA VALUE 4
2541  47F6  01      .BYTE  @001   ;DATA GROUP 7 - DATA VALUE 4 PARITY
2542
2543  47F7  00      .BYTE  @000   ;DATA GROUP 8 - DATA VALUE 1
2544  47F8  00      .BYTE  @000   ;DATA GROUP 8 - DATA VALUE 1 PARITY
2545  47F9  00      .BYTE  @000   ;DATA GROUP 8 - DATA VALUE 2
2546  47FA  00      .BYTE  @000   ;DATA GROUP 8 - DATA VALUE 2 PARITY
2547  47FB  FF      .BYTE  @377   ;DATA GROUP 8 - DATA VALUE 3
2548  47FC  01      .BYTE  @001   ;DATA GROUP 8 - DATA VALUE 3 PARITY
2549  47FD  00      .BYTE  @000   ;DATA GROUP 8 - DATA VALUE 4
2550  47FE  00      .BYTE  @000   ;DATA GROUP 8 - DATA VALUE 4 PARITY
2551
2552  47FF  FF      .BYTE  @377   ;DATA GROUP 9 - DATA VALUE 1
2553  4800  01      .BYTE  @001   ;DATA GROUP 9 - DATA VALUE 1 PARITY
2554  4801  FF      .BYTE  @377   ;DATA GROUP 9 - DATA VALUE 2
2555  4802  01      .BYTE  @001   ;DATA GROUP 9 - DATA VALUE 2 PARITY
2556  4803  00      .BYTE  @000   ;DATA GROUP 9 - DATA VALUE 3
2557  4804  00      .BYTE  @000   ;DATA GROUP 9 - DATA VALUE 3 PARITY
2558  4805  FF      .BYTE  @377   ;DATA GROUP 9 - DATA VALUE 4
2559  4806  01      .BYTE  @001   ;DATA GROUP 9 - DATA VALUE 4 PARITY
2560
2561  4807  00      .BYTE  @000   ;DATA GROUP 10 - DATA VALUE 1
2562  4808  00      .BYTE  @000   ;DATA GROUP 10 - DATA VALUE 1 PARITY
2563  4809  FF      .BYTE  @377   ;DATA GROUP 10 - DATA VALUE 2
2564  480A  01      .BYTE  @001   ;DATA GROUP 10 - DATA VALUE 2 PARITY
2565  480B  00      .BYTE  @000   ;DATA GROUP 10 - DATA VALUE 3
2566  480C  00      .BYTE  @000   ;DATA GROUP 10 - DATA VALUE 3 PARITY
2567  480D  00      .BYTE  @000   ;DATA GROUP 10 - DATA VALUE 4
2568  480E  00      .BYTE  @000   ;DATA GROUP 10 - DATA VALUE 4 PARITY
2569.
2570  480F  00      .BYTE  @000   ;DATA GROUP 11 - DATA VALUE 1
2571  4810  00      .BYTE  @000   ;DATA GROUP 11 - DATA VALUE 1 PARITY
2572  4811  FF      .BYTE  @377   ;DATA GROUP 11 - DATA VALUE 2
2573  4812  01      .BYTE  @001   ;DATA GROUP 11 - DATA VALUE 2 PARITY
2574  4813  00      .BYTE  @000   ;DATA GROUP 11 - DATA VALUE 3
2575  4814  00      .BYTE  @000   ;DATA GROUP 11 - DATA VALUE 3 PARITY
2576  4815  FF      .BYTE  @377   ;DATA GROUP 11 - DATA VALUE 4
2577  4816  01      .BYTE  @001   ;DATA GROUP 11 - DATA VALUE 4 PARITY
2578
2579  4817  00      .BYTE  @000   ;DATA GROUP 12 - DATA VALUE 1
2580  4818  00      .BYTE  @000   ;DATA GROUP 12 - DATA VALUE 1 PARITY
2581  4819  00      .BYTE  @000   ;DATA GROUP 12 - DATA VALUE 2
2582  481A  00      .BYTE  @000   ;DATA GROUP 12 - DATA VALUE 2 PARITY
```

```
2583  481B  00          .BYTE   @000        ;DATA GROUP 12 - DATA VALUE 3
2584  481C  00          .BYTE   @000        ;DATA GROUP 12 - DATA VALUE 3 PARITY
2585  481D  FF          .BYTE   @377        ;DATA GROUP 12 - DATA VALUE 4
2586  481E  01          .BYTE   @001        ;DATA GROUP 12 - DATA VALUE 4 PARITY
2587
2588  481F  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 1
2589  4820  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 1 PARITY
2590  4821  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 2
2591  4822  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 2 PARITY
2592  4823  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 3
2593  4824  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 3 PARITY
2594  4825  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 4
2595  4826  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 4 PARITY
2596
2597  4827  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 1
2598  4828  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 1 PARITY
2599  4829  FF          .BYTE   @377        ;DATA GROUP 14 - DATA VALUE 2
2600  482A  01          .BYTE   @001        ;DATA GROUP 14 - DATA VALUE 2 PARITY
2601  482B  FF          .BYTE   @377        ;DATA GROUP 14 - DATA VALUE 3
2602  482C  01          .BYTE   @001        ;DATA GROUP 14 - DATA VALUE 3 PARITY
2603  482D  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 4
2604  482E  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 4 PARITY
2605
2606  482F  00          .BYTE   @000        ;DATA GROUP 15 - DATA VALUE 1
2607  4830  00          .BYTE   @000        ;DATA GROUP 15 - DATA VALUE 1 PARITY
2608  4831  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 2
2609  4832  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 2 PARITY
2610  4833  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 3
2611  4834  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 3 PARITY
2612  4835  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 4
2613  4836  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 4 PARITY
2614
2615  4837  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 1
2616  4838  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 1 PARITY
2617  4839  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 2
2618  483A  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 2 PARITY
2619  483B  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 3
2620  483C  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 3 PARITY
2621  483D  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 4
2622  483E  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 4 PARITY
2623
2624
```

```
2626                                    .SBTTL  PROGRAM VARIABLES
2627  483F   00                         UNITMP: .BYTE   0               ;UNIT MAP
2628  4840   00                         SUNIT:  .BYTE   0               ;PORT/TU SELECT BYTE
2629  4841   00   00                    AMTMSK: .WORD   0               ;AMTIE MASK WORD
2630  4843   00   00                    INTBLTMP: .WORD 0               ;TEMP. STORAGE FOR THE INPUT TABLE POINTER
2631  4845   00   00                    INTBL:  .WORD   0               ;INPUT TABLE ADDRESS STORAGE
2632  4847   00   00                    OUTTBL: .WORD   0               ;OUTPUT TABLE ADDRESS STORAGE
2633  4849   00                         DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2634  484A   00                         DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2635  484B   00                         DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2636  484C   00                         DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2637  484D   00                         SUBCNT: .BYTE   0               ;SUBGROUP COUNTER
2638         0000                               .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK  4841           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090         BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004         BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040         BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089         CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B         CBYTL  = 008A         CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093         CDG2L  = 0092         CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D         CDVTL  = 008C         CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKAMP   468C  G        CKAM0   45C9  G       CKAM1   45E1  G       CKAM2   45F9  G
CKAM3   4611  G        CKAM4   4629  G       CKAM5   4641  G       CKAM6   4659  G
CKAM7   4671  G        CKILL   450B          CKILP   45A7  G       CKIL0   4517  G
CKIL1   4529  G        CKIL2   453B  G       CKIL3   454D  G       CKIL4   455F  G
CKIL5   4571  G        CKIL6   4583  G       CKIL7   4595  G       CKLOP  = 2815
CK5X4   45AE           CLEAR   44E0          CLKCTL= 00F0          CLOCK   4F26
CLRLP   44E4           CMC0H  = 0099         CMC0L  = 0098         CMC1H  = 0098
CMC1L  = 009A          CMC2H  = 009D         CMC2L  = 009C         CMC3H  = 009F
CMC3L  = 009E          CMINH  = 0097         CMINL  = 0096         CNTCTL= 00D7
CRCWRD= 0018           CSAVE   4F9D          CSRLH  = 0091         CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084         CTSTH  = 008F         CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080         CXINH  = 0083         CXINL  = 0082
C.     = 0001          C.AVAI= 0080          C.DP   = 0008         C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008         C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001         C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC         C.SER  = 0080         C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002         D      =%0002
DATAA   4849           DATAAP  484A          DATACT= 00D0          DATAE   484B
DATAEP  484C           DBUS    4F28          DBUSCT= 00C0          DBUSST= 00C0
DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9         DDRBIN= 0002
DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088         DDRCTL= 00DB
DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90          DIARD  = 0008
DONE1  = 0045          DONINT= 0010          DSAVE   4F9E          DSE    = 0006
DUMMX   446B  G        DUMMY   431E  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080         E      =%0003         ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041         ECCSTA= 001A          ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003          ERFLG   4F93          ERLP   = 2809
ERLPA  - 280F          ERLPB  = 2812         ERLPE  = 280C         ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F          EXIT    44DD          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080         E.CRC  = 0080         E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002          E.UNC  = 0004
FIFORD= 006A           FORMAT  4F25          FOUND   4344          FWDTST= 0061
GCRID  - 0089          GCRSET= 0002          GOODTM= 0092          H      =%0004
HLSAVE  4FA0           IE     = 0008         INTBL   4845          INTBLT  4843
INTSTA- 00E0           ITERA   4F9A          I.PWR  = 0020         I.RMPE= 0040
I5.5   = 0010          I6.5   = 0020         I7.5   = 0040         KCALL  = 005F
KCLR   - 007B          KDEP   = 003F         KENAB  = 0078         KEXAM  = 003E
KEYBRD- 00C8           KEY1   = 0078         KEY10  = 006D         KEY11  = 006E
KEY12  = 006F          KEY13  = 005C         KEY14  = 005D         KEY15  = 005E
```

```
KEY16 = 005F    KEY17 = 003C    KEY18 = 003D    KEY19 = 003E
KEY2  = 0079    KEY20 = 003F    KEY3  = 007A    KEY4  = 007B
KEY5  = 0074    KEY6  = 0075    KEY7  = 0076    KEY8  = 0077
KEY9  = 006C    KINTA = 006F    KLDAD = 003D    KN0   = 003C
KN1   = 005C    KN2   = 005D    KN3   = 005E    KN4   = 006C
KN5   = 006D    KN6   = 006E    KN7   = 0074    KN8   = 0075
KN9   = 0076    KU2   = 0079    KU3   = 007A    KU8   = 0077
L     -%0005    LBLANK= 000F    LCE     000B    LCH   = 000C
LCL   = 000D    LCP   = 000E    LC0   = 0000    LC1   = 0001
LC2   = 0002    LC3   = 0003    LC4   = 0004    LC5   = 0005
LC6   = 0006    LC7   = 0007    LC8   = 0008    LC9   = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94
LPNUM   4F92    M     =%0006    MBSEL = 00E0    MB.A  = 0008
MB.B  - 0004    MEMTOP= 4FFF    MINUS = 000A    MM    = 8000
MSE   = 0008    MSGN  = 0014    MTACLR= 0000    MT.ARA= 0020
MT.CPE- 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT- 0080    M.DEM = 0020    M.EBL = 0010    M.EXC = 0008
M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010    M.OCC = 0020
M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5  = 0001    M6.5  = 0002    M7.5  = 0004    M8950E  4693
M8950I  471F    M89500  47BF    NOTCAP= 0088    OKAY  = 00FF
OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040    OUT     4692
OUTILL  45AD    OUTTBL  4847    PADCNT= 00D5    PADCRC= 0080
PDIAG = 0048    PEID  = 008A    PENAB = 004C    PESET = 0001
PL    = 0081    PRDD  = 004C    PRENF = 009C    PS    = 0082
PSTAT = 0048    PSW   =%0009    P.AMTP= 0001    P.BCTC= 0040
P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040    P.LWR = 0020
P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002    P.RPOE= 0020
P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080
P.STAT= 0002    P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010
P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001    P.WPEN= 0010
P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004
P.5VOK= 0002    QUE   = 2818    QUEM  = 281E    RAMT  = 0010
RARA  = 0006    RARAI = 0004    RCHBD0= 0048    RCHBD1= 0047
RCHOK = 0046    RCHTST= 000C    RCLRT = 000D    RCMD  = 0008
RCMLP = 0003    RCONT = 0080    RDATA = 0017    RDCLK = 0010
RDON  = 0011    READG = 0007    REND  = 0014    REQST = 2806
RESCHR= 00D1    REVTST= 0064    REWIND= 0004    RFIFOL= 0008
RGCLK = 0002    RGCRI - 0003    RIBG  = 0001    RILL  = 0012
RINST = 000C    RMCTST= 0008    RMK2  = 0013    RNOP  = 0000
RPATH = 0001    RPBAD = 0044    RPCHI = 0001    RPCLK = 0003
RPCTL = 0009    RPEI  = 0002    RPFAIL= 0000    RPF1  = 009D
RPF2  = 009E    RPOK  = 0043    RPOSTN= 0016    RPSTA = 0015
RRCMT = 000A    RSTAT = 0002    RTIEB = 000A    RTIER = 0030
RTM   = 0005    RUNKI = 0009    RUPTST= 005E    RWDUNL= 0005
R.AMT = 0001    R.BOP = 0008    R.DATA= 0040    R.DON = 0002
```

RPM4 - READ PATH MICRO CONTROLLER PART #4 CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:04 PAGE 1-35 SEQ 0822
RPM4.M80 SYMBOL TABLE

```
R.DRDY= 0010    R.END = 0010    R.ILL = 0004    R.JVOK= 0004
R.MK2 = 0008    R.PLOD= 0008    R.PL00= 0010    R.PL01= 0020
R.POST= 0020    R.STNM= 0002    R.STOP= 0004    R.STPC= 0001
R.TBJN= 0080    R.TSTD= 0040    R.VOK = 0080    R00H  = 0081
R00L  = 0080    R01H  = 0083    R01L  = 0082    R02H  = 0085
R02L  = 0084    R03H  = 0087    R03L  = 0086    R04H  = 0089
R04L  = 0088    R05H  = 008B    R05L  = 008A    R06H  = 008D
R06L  = 008C    R07H  = 008F    R07L  = 008E    R10H  = 0091
R10L  = 0090    R11H  = 0093    R11L  = 0092    R12H  = 0095
R12L  = 0094    R13H  = 0097    R13L  = 0096    R14H  = 0099
R14L  = 0098    R15H  = 009B    R15L  = 009A    R16H  = 009D
R16L  = 009C    R17H  = 009F    R17L  = 009E    R7.5  = 0010
SELCLR= 0000    SETATA= 00A1    SID   = 0080    SOD   = 0080
SOE   = 0040    SP    =%0008    SSCLK = 0040    SSTEP = 0005
STACK = 4FFF    STATRM= 4F20    STPCT   4F20    STRSP = 5000
SUBCNT  484D    SUNIT   4840    TADR00= 0080    TADR01= 0081
TADR02= 0082    TADR03= 0083    TADR04= 0084    TADR05= 0085
TADR06= 0086    TADR07= 0087    TADR10= 0088    TADR11= 0089
TADR12= 008A    TADR13= 008B    TAMT  = 0044    TASEL = 0080
TCMD  = C040    TC.FWD= 0040    TC.INH= 0008    TC.LWR= 0004
TC.REV= 0020    TC.WRT= 0010    TEMP    4F99    TEST1   4300
TEST2   43E0    TMF   = 0099    TMRDY = 0040    TRKENA= 00D2
TSET  = 2803    TSTEND= 2818    TSTS  = 0040    TST01A  43B8
TST01C  4377    TST01X  4351    TST02A  4474    TST02B  447C
TST02C  441A    TST02X  43E5    TUSEL0= 00D1    TUSEL1= 00D2
TU78  = 0010    T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004
T.EOT = 0002    T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020
T.PES = 0008    T.PSBJ= 0020    T.PS0J= 0008    T.PS1J= 0010
T.RDY = 0080    T.RDYO= 0040    T.RWD = 0010    T.SCLK= 0002
UIBG  = 00A1    UNITMP  483F    UNITSL  4327    VALFC   4F98
VALTB   4F95    VELTST= 005B    WDR.P = 0010    WMCCTL= 00D3
WMCERR= 00DA    WMCSTA= 00D0    WRTCLK= 0000    WRTDAT= 00D3
W.ACRC= 0004    W.CRC = 0008    W.DIAG= 0002    W.DONN= 0040
W.ECC = 0010    W.ENAB= 0080    W.ERR = 0020    W.FMT = 0070
W.GCR = 0010    W.LEFT= 0004    W.ONES= 0020    W.RESI= 0002
W.REV = 0004    W.ROME= 0010    W.RST = 0001    W.SKIP= 000F
W.WRIT= 0008    W.XFER= 0020    X     =%000A    X.DONN= 0080
X.ENAB= 0040    X.PEPE= 0002    X.ROME= 0001    X.WCLK= 0001
Y     -%0008    .     = 484E
```

ERRORS DETECTED: 0

*RPM4.A78/PTP,RPM4=NLIST,PARAM,MACRO,LIST,RPM4
RUN-TIME: 4 6 0 SECONDS
CORE USED: 10K

```
1329                                    .TITLE  RPM5 - READ PATH MICRO CONTROLLER PART #5
1330                                    .SBTTL  TEST 01 - MARK TWO DETECT
1331                                    ;ID     RPM5-READ PATH MICRO CONTROLLER PART #5
1332  4300                              ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1333                                    ;*MARK TWO DETECT
1334  4300                              SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1335                                    ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
1336                                    ;*AS WELL AS THE READ CHANNELS ABILITY TO DETECT A MARK TWO TRANSLATION.
1337  4300                              SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1338                                    ;*BGNTST
1339                                    ;*      SET NORMAL READ PATH CLOCK
1340                                    ;*      CALL SUBROUTINE CLEAR
1341                                    ;*      SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1342                                    ;*      CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1343                                    ;*      DELAY 50 MICROSECONDS
1344                                    ;*      CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1345                                    ;*      CLEAR THE TIE BUS
1346                                    ;*      SET UP INPUT TABLE POINTER
1347                                    ;*      ENABLE THE READ PATH CLOCK
1348                                    ;*      SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
1349                                    ;*      CLOCK THE FIFO
1350                                    ;*      ISSUE CLEAR ALL COMMAND
1351                                    ;*      WAIT
1352                                    ;*      STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1353                                    ;*      INIT THE LOOP COUNT TO 5
1354                                    ;*      BGNDO
1355                                    ;*      :     MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
1356                                    ;*      :     INCREMENT THE INPUT TABLE POINTER
1357                                    ;*      :     MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
1358                                    ;*      :     CLOCK THE DATA INTO THE FIFO
1359                                    ;*      :     DECREMENT THE LOOP COUNT
1360                                    ;*      :     INCREMENT THE INPUT TABLE POINTER
136*                                    ;*      :     DO UNTIL LOOP COUNT = 0
1362                                    ;*      ENDDO
1363                                    ;*      ISSUE DIAGNOSTIC READ COMMAND
1364                                    ;*      SET UP WATCHDOG TIMER COUNT
1365                                    ;*      BGNDO
1366                                    ;*      :     SINGLE STEP THE READ PATH
1367                                    ;*      :     DECREMENT THE WATCHDOG TIMER
1368                                    ;*      :     DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1369                                    ;*      ENDDO
1370                                    ;*      IF WATCHDOG TIMER=0
1371                                    ;*      :     THEN-ERROR
1372                                    ;*      :     ELSE-CONTINUE
1373                                    ;*      ENDIF
```

```
1374                                    ;*    SINGLE STEP THE READ PATH
1375                                    ;*    CALL SUBROUTINE CKMK
1376                                    ;*    DO UNTIL END OF INPUT DATA TABLE
1377                                    ;*ENDTST
1378  4300                              SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1379                                    ;*RPM5 MICRO TEST 01
1380                                    ;*RPM5 MICRO ERROR 01
1381                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
1382                                    ;*M8950'S, M8953
1383                                    ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1384                                    ;*FATAL ERROR - TEST ABORTED
1385                                    ;*
1386                                    ;*RPM5 MICRO TEST 01
1387                                    ;*RPM5 MICRO ERROR 02
1388                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
1389                                    ;*M8950'S, M8953
1390                                    ;*CHANNEL 0 NOT MARK TWO
1391                                    ;*
1392                                    ;*RPM5 MICRO TEST 01
1393                                    ;*RPM5 MICRO ERROR 03
1394                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
1395                                    ;*M8950'S, M8953
1396                                    ;*CHANNEL 1 NOT MARK TWO
1397                                    ;*
1398                                    ;*RPM5 MICRO TEST 01
1399                                    ;*RPM4 MICRO ERROR 04
1400                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
1401                                    ;*M8950'S, M8953
1402                                    ;*CHANNEL 2 NOT MARK TWO
1403                                    ;*
1404                                    ;*RPM5 MICRO TEST 01
1405                                    ;*RPM5 MICRO ERROR 05
1406                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
1407                                    ;*M8950'S, M8953
1408                                    ;*CHANNEL 3 NOT MARK TWO
1409                                    ;*
1410                                    ;*RPM5 MICRO TEST 01
1411                                    ;*RPM5 MICRO ERROR 06
1412                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
1413                                    ;*M8950'S, M8953
1414                                    ;*CHANNEL 4 NOT MARK TWO
1415                                    ;*
1416                                    ;*RPM5 MICRO TEST 01
1417                                    ;*RPM5 MICRO ERROR 07
1418                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
1419                                    ;*M8950'S, M8953
1420                                    ;*CHANNEL 5 NOT MARK TWO
1421                                    ;*
1422                                    ;*RPM5 MICRO TEST 01
1423                                    ;*RPM5 MICRO ERROR 10
1424                                    ;*RPM5-READ PATH-MARK TWO DETECT TEST
```

```
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 6 NOT MARK TWO
                                        ;*
                                        ;*RPM5 MICRO TEST 01
                                        ;*RPM5 MICRO ERROR 11
                                        ;*RPM5-READ PATH-MARK TWO DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 7 NOT MARK TWO
                                        ;*
                                        ;*RPM5 MICRO TEST 01
                                        ;*RPM5 MICRO ERROR 12
                                        ;*RPM5-READ PATH-MARK TWO DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P NOT MARK TWO
                                        S
 (1)                                    ; ************************************************************************************

                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
                                        ;%RPM5-READ PATH-MARK TWO DETECT TEST
                                        ;&M8950'S, M8953
                                                REQ     @7,0,0,0,0
 (1)  4305  CD  06  28       18.0                CALL    REQST
 (1)  4308  00                          .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                      .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                      .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                          .BYTE   0               ;DATA COMPARE FLAG IF -1
 (1)  430E  07                          .BYTE   @7              ;REQUEST CODE
                                                RIN     R12L
 (1)  430F  DB  94            10.0              IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                 4.0              MOV     A,A             ;RETRY LINK
                                                STA     UNITMP
                                                ANA     A
                                                JNZ     UNITSL
                                                ERR     EXIT,DUMMY
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28       18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                          .BYTE
 (1)  431E  CD  15  28       18.0        DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  0B  45       10.0                JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;>FATAL ERROR - TEST ABORTED
                                                JMP     EXIT

 4327  06  00                 7.0  UNITSL: MVI     B,@0            ;LOAD THE BASE UNIT NUMBER
 4329  3A  78  47            13.0          LDA     UNITMP          ;DID THE USER SPECIFY TU
 432C  E6  01                 7.0          ANI     @01             ;PORT 0?
 432E  C2  44  43            10.0          JNZ     FOUND           ;YES-GO USE PORT #0
 4331  04                     4.0          INR     B               ;NO-UPDATE POINTER TO PORT #1
 4332  3A  78  47            13.0          LDA     UNITMP          ;DID THE USER SPECIFY TU
 4335  E6  02                 7.0          ANI     @02             ;PORT 1?
```

```
1461  4337  C2  44  43    10.0          JNZ   FOUND          ;YES-GO USE PORT #1
1462  433A  04            4.0          INR   B              ;NO-UPDATE POINTER TO PORT #2
1463  433B  3A  78  47    13.0          LDA   UNITMP         ;DID THE USER SPECIFY TU
1464  433E  E6  04         7.0          ANI   @04            ;PORT 2?
1465  4340  C2  44  43    10.0          JNZ   FOUND          ;YES-GO USE PORT #2
1466  4343  04            4.0          INR   B              ;NO-ASSUME PORT #3
1467  4344  CD  0E  45    18.0  FOUND:  CALL  CLEAR          ;CLEAR ALL THE TU PORTS
1468  4347  DB  E0        10.0          IN    INTSTA         ;GET THE MASS BUS SELECT BIT
1469  4349  E6  80         7.0          ANI   BIT7           ;
1470  434B  B0            4.0          ORA   B              ;OR IN THE TU PORT SELECT BITS
1471  434C  D3  E0        10.0          OUT   MBSEL          ;SELECT THE MASS BUS AND TU PORTS
1472  434E  32  79  47    13.0          STA   SUNIT          ;STORE THE UNIT NUMBER FOR LATER USE
1473
1474  4351  3E  10         7.0  TST01X: MVI   A,RDCLK       ;SET NORMAL READ PATH CLOCKS
1475  4353  D3  F0        10.0          OUT   CLKCTL
1476
1477  4355  3E  04         7.0          MVI   A,P.RPEN               ;ENABLE THE READ PATH FROM THE TU PORT
1478  4357  D3  4C        10.0          OUT   PENAB          ;
1479  4359  3E  10         7.0          MVI   A,W.GCR        ;SET GCR MODE
1480  435B  D3  D3        10.0          OUT   WMCCTL
1481  435D  3E  60         7.0          MVI   A,P.LWR!P.LCS  ;SET THE PORT CONTROL TO
1482  435F  D3  48        10.0          OUT   PDIAG          ;LCS, LWR AND DPEN
1483  4361  AF            4.0          XRA   A
1484  4362  D3  44        10.0          OUT   TAMT           ;
1485  4364  3E  0A         7.0          MVI   A,10           ;DELAY
1486  4366  3D            4.0  1$:     DCR   A              ;50
1487  4367  C2  66  43    10.0          JNZ   1$             ;MICROSECONDS
1488
1489  436A  D3  D2        10.0          OUT   TRKENA         ;CLEAR ALL TRACKS FROM
1490  436C  D3  D2        10.0          OUT   TRKENA         ;THE TRANSLATOR
1491
1492  436E  AF            4.0          XRA   A              ;CLEAR THE TIE BUS
1493  436F  D3  0A        10.0          OUT   RTIEB
1494
1495  4371  21  64  47    10.0          LXI   H,M8950MK     ;SET UP POINTER TO INPUT DATA TABLE
1496  4374  22  7E  47    16.0          SHLD  INTBL          ;SAVE IT
1497  4377  3E  A8         7.0  TST01C: MVI   A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1498                                                        ;TIE BUS JAM AND PLO DISABLE
1499  4379  D3  09        10.0          OUT   RPCTL
1500  437B  D3  08        10.0          OUT   RFIFOL         ;CLOCK THE FIFO
1501  437D  3E  0D         7.0          MVI   A,RCLRT        ;ISSUE CLEAR ALL COMMAND
1502  437F  D3  0B        10.0          OUT   RCMD
1503
1504  4381  00            4.0          NOP                  ;WAIT
1505  4382  00            4.0          NOP
1506  4383  00            4.0          NOP
1507  4384  00            4.0          NOP
1508
1509  4385  3E  A9         7.0          MVI   A,R.PLO1!R.STPC.R.PLOD!R.TBJN   ;STOP THE READ PATH
1510                                                                ;SET TIE BUS JAM AND PLO DISABLE
1511  4387  D3  09        10.0          OUT   RPCTL
1512
1513  4389  0E  02         7.0          MVI   C,2            ;SET UP THE FIFO LOAD COUNTER
1514  438B  2A  7E  47    16.0  3$:     LHLD  INTBL          ;GET POINTER TO INPUT DATA TABLE
```

```
1515  438E  06  05          7.0          MVI   B,@5              ;SET UP LOOP COUNT
1516  4390  7E              7.0   1$:    MOV   A,M               ;GET A DATA BYTE
1517  4391  D3  40         10.0          OUT   TCMD              ;STORE DATA IN COMMAND ADDRESS
1518  4393  23              6.0          INX   H                 ;POINT TO DATA PARITY
1519  4394  7E              7.0          MOV   A,M               ;GET THE DATA PARITY
1520  4395  07              4.0          RLC                     ;POSITION FOR OUTPUT
1521  4396  F6  60          7.0          ORI   P.LWR!P.LCS       ;OR IN CONTROL BITS
1522  4398  D3  48         10.0          OUT   PDIAG             ;OUTPUT THE DATA PARITY
1523  439A  D3  08         10.0          OUT   RFIFOL            ;CLOCK DATA INTO THE FIFO'S
1524
1525  439C  23              6.0          INX   H                 ;UPDATE THE TABLE POINTER
1526  439D  05              4.0          DCR   B                 ;DECREMENT LOOP COUNT
1527  439E  C2  90  43      10.0          JNZ   1$                ;DO UNTIL LOOP COUNT = 0
1528
1529  43A1  0D              4.0          DCR   C                 ;DECREMENT THE FIFO LOAD LOOP COUNTER
1530  43A2  C2  88  43      10.0          JNZ   3$                ;CONTINUE UNTIL ZERO
1531                                  ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1532
1533  43A5  22  7E  47      16.0          SHLD  INTBL             ;SAVE THE INPUT DATA TABLE POINTER
1534  43A8  3E  0B          7.0          MVI   A,DIARD           ;LOAD THE DIAGNOSTIC READ
1535  43AA  D3  0B         10.0          OUT   RCMD              ;COMMAND
1536  43AC  11  01  00     10.0          LXI   D,1               ;SET WATCH DOG INCREMENT
1537  43AF  21  A8  FD     10.0          LXI   H,-600            ;SET WATCH DOG COUNT TO 600
1538  43B2  D3  0C         10.0   2$:    OUT   RINST             ;STEP THE READ PATH
1539  43B4  DB  01         10.0          IN    RPCHI             ;GET THE READ PATH STATUS
1540  43B6  E6  10          7.0          ANI   R.DRDY            ;DATA READY SET?
1541  43B8  C2  BF  43      10.0          JNZ   TST01A            ;YES - GO CHECK
1542  43BB  19              10.0          DAD   D                 ;WATCH DOG TIMEOUT?
1543  43BC  D2  B2  43      10.0          JNC   2$                ;NO-CONTINUE
1544  43BF  D3  0C         10.0  TST01A: OUT   RINST             ;SINGLE STEP THE READ PATH
1545  43C1  CD  39  45      18.0          CALL  CKMK              ;CHECK FOR FAILING READ CHANNELS
1546  43C4  DA  51  43      10.0          JC    TST01X            ;LOOP ON ERROR
1547  43C7                                ENDTST TST01X
 (1)                              ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43C7                          REQ   7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43C7  CD  06  28     18.0                          CALL    REQST
 (2)  43CA  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  43CB  00  00                                       .WORD                   ;SYSTEM "*" COUNT
 (2)  43CD  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43CF  00                                           .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  43D0  07                                           .BYTE   7               ;REQUEST CODE
 (1)  43D1  3A  9A  4F     13.0          LDA   ITERA             ;GET ITERATION COUNT
 (1)  43D4  3D              4.0          DCR   A                 ;DOWNCOUNT
 (1)  43D5  32  9A  4F     13.0          STA   ITERA             ;SAVE COUNT
 (1)  43D8  F2  51  43     10.0          JP    TST01X            ;DO TEST UNTIL TILL = 0
1548  43DB  CD  0E  45     18.0          CALL  CLEAR             ;CLEAR ALL TU PORTS
```

```
 1550                                   .SBTTL  TEST 02 - END MARK DETECT TEST
 1551  43DE                            ST
  (1)                                   ;*****************************************************************************
  (1)                                   ;*TEST TITLE
  (1)                                   ;*---- -----
 1552                                   ;*END MARK DETECT TEST
 1553  43DE                            SD
  (1)                                   ;*****************************************************************************
  (1)                                   ;*DESCRIPTION
  (1)                                   ;*-----------
 1554                                   ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
 1555                                   ;*AS WELL AS THE READ CHANNELS ABILITY TO DETECT AN END MARK TRANSLATION.
 1556  43DE                            SP
  (1)                                   ;*****************************************************************************
  (1)                                   ;*PROCEDURE
  (1)                                   ;*---------
 1557                                   ;*BGNTST
 1558                                   ;*     SET NORMAL READ PATH CLOCK
 1559                                   ;*     CALL SUBROUTINE CLEAR
 1560                                   ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
 1561                                   ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
 1562                                   ;*     DELAY 50 MICROSECONDS
 1563                                   ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
 1564                                   ;*     CLEAR THE TIE BUS
 1565                                   ;*     SET UP INPUT TABLE POINTER
 1566                                   ;*     ENABLE THE READ PATH CLOCK
 1567                                   ;*     SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
 1568                                   ;*     CLOCK THE FIFO
 1569                                   ;*     ISSUE CLEAR ALL COMMAND
 1570                                   ;*     WAIT
 1571                                   ;*     STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
 1572                                   ;*     INIT THE LOOP COUNT TO 5
 1573                                   ;*     BGNDO
 1574                                   ;*     :     MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
 1575                                   ;*     :     INCREMENT THE INPUT TABLE POINTER
 1576                                   ;*     :     MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
 1577                                   ;*     :     CLOCK THE DATA INTO THE FIFO
 1578                                   ;*     :     DECREMENT THE LOOP COUNT
 1579                                   ;*     :     INCREMENT THE INPUT TABLE POINTER
 1580                                   ;*     :     DO UNTIL LOOP COUNT = 0
 1581                                   ;*     ENDDO
 1582                                   ;*     ISSUE DIAGNOSTIC READ COMMAND
 1583                                   ;*     SET UP WATCHDOG TIMER COUNT
 1584                                   ;*     BGNDO
 1585                                   ;*     :     SINGLE STEP THE READ PATH
 1586                                   ;*     :     DECREMENT THE WATCHDOG TIMER
 1587                                   ;*     :     DO UNTIL WATCHDOG TIMER - 0 OR DATA READY SETS
 1588                                   ;*     ENDDO
 1589                                   ;*     IF WATCHDOG TIMER=0
 1590                                   ;*     :     THEN-ERROR
 1591                                   ;*     :     ELSE-CONTINUE
 1592                                   ;*     ENDIF
 1593                                   ;*     SINGLE STEP THE READ PATH
 1594                                   ;*     CALL SUBROUTINE CKEMK
```

```
                                        ;*    DO UNTIL END OF INPUT DATA TABLE
                                        ;*ENDTST
1597  43DE                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 13
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 0 NOT END MARK
                                        ;*
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 14
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 1 NOT END .MARK
                                        ;*
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 15
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 2 NOT END MARK
                                        ;*
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 16
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 3 NOT END MARK
                                        ;*
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 17
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 4 NOT END MARK
                                        ;*
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 20
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 5 NOT END MARK
                                        ;*
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 21
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 6 NOT END MARK
                                        ;*
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 22
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 7 NOT END MARK
                                        ;*
```

```
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 23
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P NOT END MARK
                                        S
 (1)                                    ; ********************************************************************************
                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1) 43DE 3E 02          7.0                    MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1) 43E0 CD 03 28      18.0                    CALL    TSET            ;SETUP THE TEST
                                        ;%RPM5-READ PATH-END MARK DETECT TEST
                                        ;&M8950'S, M8953

 43E3 3A 79 47          13.0           LDA     SUNIT
 43E6 D3 E0             10.0           OUT     MBSEL
 43E8 3E 10              7.0  TST02X:  MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
 43EA D3 F0             10.0           OUT     CLKCTL

 43EC 3E 04              7.0           MVI     A,P.RPEN                ;ENABLE THE READ PATH FROM THE TU PORT
 43EE D3 4C             10.0           OUT     PENAB           ;
 43F0 3E 10              7.0           MVI     A,W.GCR         ;SET GCR MODE
 43F2 D3 D3             10.0           OUT     WMCCTL
 43F4 3E 60              7.0           MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
 43F6 D3 48             10.0           OUT     PDIAG           ;LCS, LWR AND DPEN
 43F8 AF                 4.0           XRA     A
 43F9 D3 44             10.0           OUT     TAMT            ;
 43FB 3E 0A              7.0           MVI     A,10            ;DELAY
 43FD 3D                 4.0  1$:      DCR     A               ;50
 43FE C2 FD 43          10.0           JNZ     1$              ;MICROSECONDS

 4401 D3 D2             10.0           OUT     TRKENA          ;CLEAR ALL TRACKS FROM
 4403 D3 D2             10.0           OUT     TRKENA          ;THE TRANSLATOR

 4405 AF                 4.0           XRA     A               ;CLEAR THE TIE BUS
 4406 D3 0A             10.0           OUT     RTIEB

 4408 21 6E 47          10.0           LXI     H,M8950EM       ;SET UP POINTER TO INPUT DATA TABLE
 440B 22 7E 47          16.0           SHLD    INTBL           ;SAVE IT
 440E 3E A8              7.0  TST02C:  MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
                                                                ;TIE BUS JAM AND PLO DISABLE
 4410 D3 08             10.0           OUT     RFIFOL          ;CLOCK THE FIFO
 4412 D3 09             10.0           OUT     RPCTL
 4414 3E 0D              7.0           MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
 4416 D3 0B             10.0           OUT     RCMD

 4418 00                 4.0           NOP                     ;WAIT
 4419 00                 4.0           NOP
 441A 00                 4.0           NOP
 441B 00                 4.0           NOP

 441C 3E A9              7.0           MVI     A,R.PLO1!R.STPC!R.PLOD.R.TBJN   ;STOP THE READ PATH
                                                                ;SET TIE BUS JAM AND PLO DISABLE
 441E D3 09             10.0           OUT     RPCTL
```

```
1697
1698  4420  0E  02          7.0          MVI   C,2              ;SET UP THE FIFO LOAD COUNTER
1699  4422  2A  7E  47      16.0  3$:     LHLD  INTBL            ;GET POINTER TO INPUT DATA TABLE
1700  4425  06  05          7.0          MVI   B,@5             ;SET UP LOOP COUNT
1701  4427  7E              7.0  1$:     MOV   A,M              ;GET A DATA BYTE
1702  4428  D3  40         10.0          OUT   TCMD             ;STORE DATA IN COMMAND ADDRESS
1703  442A  23              6.0          INX   H                ;POINT TO DATA PARITY
1704  442B  7E              7.0          MOV   A,M              ;GET THE DATA PARITY
1705  442C  07              4.0          RLC                    ;POSITION FOR OUTPUT
1706  442D  F6  60          7.0          ORI   P.LWR.P.LCS  ;OR IN CONTROL BITS
1707  442F  D3  48         10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
1708  4431  D3  08         10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFO'S
1709
1710  4433  23              6.0          INX   H                ;UPDATE THE TABLE POINTER
1711  4434  05              4.0          DCR   B                ;DECREMENT LOOP COUNT
1712  4435  C2  27  44      10.0          JNZ   1$               ;DO UNTIL LOOP COUNT = 0
1713
1714  4438  0D              4.0          DCR   C                ;DECREMENT THE FIFO LOAD LOOP COUNTER
1715  4439  C2  22  44      10.0          JNZ   3$               ;CONTINUE UNTIL ZERO
1716                                  ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1717
1718  443C  22  7E  47      16.0          SHLD  INTBL            ;SAVE THE INPUT DATA TABLE POINTER
1719  443F  3E  0B          7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
1720  4441  D3  0B         10.0          OUT   RCMD             ;COMMAND
1721  4443  11  01  00      10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
1722  4446  21  A8  FD      10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
1723  4449  D3  0C         10.0  2$:     OUT   RINST            ;STEP THE READ PATH
1724  444B  DB  01         10.0          IN    RPCHI            ;GET THE READ PATH STATUS
1725  444D  E6  10          7.0          ANI   R.DRDY           ;DATA READY SET?
1726  444F  C2  56  44      10.0          JNZ   TST02A           ;YES - GO CHECK
1727  4452  19              10.0          DAD   D                ;WATCH DOG TIMEOUT?
1728  4453  D2  49  44      10.0          JNC   2$               ;NO-CONTINUE
1729  4456  D3  0C         10.0  TST02A: OUT   RINST            ;SINGLE STEP THE READ PATH
1730  4458  CD  DC  45      18.0          CALL  CKEMK            ;CHECK FOR FAILING READ CHANNELS
1731  445B  DA  E8  43      10.0          JC    TST02X           ;LOOP ON ERROR
1732  445E                                ENDTST TST02X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  445E                          REQ   7                      ;FAKE CALL TO KEEP TEST ALIVE
 (2)  445E  CD  06  28      18.0                          CALL    REQST
 (2)  4461  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  4462  00  00                                        .WORD                   ;SYSTEM '*' COUNT
 (2)  4464  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4466  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4467  07                                            .BYTE   7               ;REQUEST CODE
 (1)  4468  3A  9A  4F      13.0          LDA   ITERA            ;GET ITERATION COUNT
 (1)  446B  3D              4.0          DCR   A                ;DOWNCOUNT
 (1)  446C  32  9A  4F      13.0          STA   ITERA            ;SAVE COUNT
 (1)  446F  F2  E8  43      10.0          JP    TST02X           ;DO TEST UNTIL TILL = 0
1733  4472  CD  0E  45      18.0          CALL  CLEAR            ;CLEAR ALL TU PORTS
```

```
1735                                .SBTTL  TEST 03 - FIFO DEPTH TEST
1736  4475                          ST
 (1)                                ; ***************************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1737                                ;*FIFO DEPTH TEST
1738  4475                          SD
 (1)                                ; ***************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1739                                ;*THIS TEST IS DESIGNED TO PROVIDE A COMPLETE TEST OF THE FIFO DEPTH TO
1740                                ;*INSURE THAT THE READ CHANNELS CAN RUN WITH THEIR SPECIFIED SKEW VALUE.
1741  4475                          SP
 (1)                                ; ***************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1742                                ;*BGNTST
1743                                ;*  SET NORMAL READ PATH CLOCKS
1744                                ;*  CLEAR THE EXPECTED DATA AND EXPECTED DATA PARITY
1745                                ;*  SET PLO BYPASS MODE 3,TIE BUS JAM AND PLO DISABLE
1746                                ;*  CLOCK THE FIFO
1747                                ;*  ISSUE CLEAR ALL FIFO'S COMMAND
1748                                ;*  WAIT
1749                                ;*  BGNDO
1750                                ;*  :  SET UP FIFO DATA OF ZERO, TIE BUS JAM AND PLO DISABLE
1751                                ;*  :  CLOCK THE DATA INTO THE FIFO
1752                                ;*  :  SET UP FIFO DATA OF ONE, TIE BUS JAM AND PLO DISABLE
1753                                ;*  :  CLOCK THE DATA INTO THE FIFO
1754                                ;*  :  DECREMENT THE LOOP COUNT
1755                                ;*  :  DO UNTIL THE LOOP COUNT=0
1756                                ;*  ENDDO
1757                                ;*  CLEAR THE LOOP COUNT
1758                                ;*  BGNDO
1759                                ;*  :  PUT THE TM78 IN PE MODE
1760                                ;*  :  ISSUE THE FIFO READ COMMAND
1761                                ;*  :  LET THE RMC RUN, SET TIE BUS JAM AND PLO DISABLE
1762                                ;*  :  WAIT
1763                                ;*  :  STOP THE RMC, SET TIE BUS JAM AND PLO DISABLE
1764                                ;*  :  INPUT THE FIFO DATA
1765                                ;*  :  INPUT THE FIFO DATA PARITY
1766                                ;*  :  CALL SUBROUTINE CKAMT
1767                                ;*  :  INCREMENT THE LOOP COUNT
1768                                ;*  :  DO UNTIL THE LOOP COUNT=62(10)
1769                                ;*  ENDDO
1770                                ;*ENDTST
1771  4475                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1772                                ;*RPM5 MICRO TEST 03
1773                                ;*RPM5 MICRO ERROR 24
1774                                ;*RPM5-FIFO DEPTH TEST
1775                                ;*M8950, M8953
1776                                ;*CHANNEL 0 FAILED
```

```
1777                    ;*ACTUAL = NNNN
1778                    ;*EXPECTED = NNNN
1779                    ;*
1780                    ;*RPM5 MICRO TEST 03
1781                    ;*RPM5 MICRO ERROR 25
1782                    ;*RPM5-FIFO DEPTH TEST
1783                    ;*M8950, M8953
1784                    ;*CHANNEL 1 FAILED
1785                    ;*ACTUAL = NNNN
1786                    ;*EXPECTED = NNNN
1787                    ;*
1788                    ;*RPM5 MICRO TEST 03
1789                    ;*RPM5 MICRO ERROR 26
1790                    ;*RPM5-FIFO DEPTH TEST
1791                    ;*M8950, M8953
1792                    ;*CHANNEL 2 FAILED
1793                    ;*ACTUAL = NNNN
1794                    ;*EXPECTED = NNNN
1795                    ;*
1796                    ;*RPM5 MICRO TEST 03
1797                    ;*RPM5 MICRO ERROR 27
1798                    ;*RPM5-FIFO DEPTH TEST
1799                    ;*M8950, M8953
1800                    ;*CHANNEL 3 FAILED
1801                    ;*ACTUAL = NNNN
1802                    ;*EXPECTED = NNNN
1803                    ;*
1804                    ;*RPM5 MICRO TEST 03
1805                    ;*RPM5 MICRO ERROR 30
1806                    ;*RPM5-FIFO DEPTH TEST
1807                    ;*M8950, M8953
1808                    ;*CHANNEL 4 FAILED
1809                    ;*ACTUAL = NNNN
1810                    ;*EXPECTED = NNNN
1811                    ;*
1812                    ;*RPM5 MICRO TEST 03
1813                    ;*RPM5 MICRO ERROR 31
1814                    ;*RPM5-FIFO DEPTH TEST
1815                    ;*M8950, M8953
1816                    ;*CHANNEL 5 FAILED
1817                    ;*ACTUAL = NNNN
1818                    ;*EXPECTED = NNNN
1819                    ;*
1820                    ;*RPM5 MICRO TEST 03
1821                    ;*RPM5 MICRO ERROR 32
1822                    ;*RPM5-FIFO DEPTH TEST
1823                    ;*M8950, M8953
1824                    ;*CHANNEL 6 FAILED
1825                    ;*ACTUAL = NNNN
1826                    ;*EXPECTED = NNNN
1827                    ;*
1828                    ;*RPM5 MICRO TEST 03
1829                    ;*RPM5 MICRO ERROR 33
1830                    ;*RPM5-FIFO DEPTH TEST
```

```
                                        ;*M8950, M8953
                                        ;*CHANNEL 7 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM5 MICRO TEST 03
                                        ;*RPM5 MICRO ERROR 34
                                        ;*RPM5-FIFO DEPTH TEST
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;
4475                                    $
                                        ; ***************************************************************************************
4475                                    TEST3:  TESTX   @3              ;INITIALIZE THE TEST
4475  3E  03              7.0                       MVI     A,@3          ;DEFINE THE TEST NUMBER
4477  CD  03  28         18.0                       CALL    TSET          ;SETUP THE TEST
                                        ;%RPM5-FIFO DEPTH TEST
                                        ;&M8950, M8953

447A  3E  10              7.0  TST03X: MVI     A,RDCLK          ;SET NORMAL READ PATH CLOCKS
447C  D3  F0             10.0         OUT     CLKCTL
447E  AF                  4.0         XRA     A                ;CLEAR
447F  32  87  47         13.0         STA     PASCNT           ;PASS COUNT
4482  32  84  47         13.0         STA     DATAE            ;EXPECTED DATA
4485  32  85  47         13.0         STA     DATAEP           ;EXPECTED DATA PARITY
4488                                   ROUT    R05H             ;CAS REGISTER 05 HIGH
4488  D3  88             10.0                 OUT     R05H             ;WRITE AC INTO R05H
448A  7F                  4.0                 MOV     A,A              ;RETRY LINK
448B  3E  B8              7.0         MVI     A,R.PLO1!R.PLO0!R.PLOD!R.TBJN   ;SET PLO BYPASS MODE 3
                                                                       ;AND TIE BUS JAM AND PLO DISABLE
448D  D3  09             10.0         OUT     RPCTL            ;
448F  D3  08             10.0         OUT     RFIFOL           ;CLOCK THE FIFO
4491  3E  0D              7.0         MVI     A,RCLRT          ;CLEAR THE FIFO'S
4493  D3  08             10.0         OUT     RCMD
4495  00                  4.0         NOP                      ;DELAY
4496  00                  4.0         NOP                      ;
4497  00                  4.0         NOP                      ;
4498  00                  4.0         NOP                      ;
4499  06  20              7.0         MVI     B,32             ;LOAD THE LOOP COUNT
449B  3E  B9              7.0  2$:    MVI     A,R.PLO0!R.PLO1!R.STPC!R.PLOD!R.TBJN  ;LOAD FIFO DATA OF ZERO
                                                                       ;AND SET TIE BUS JAM AND PLO DISABLE
449D  D3  09             10.0         OUT     RPCTL
449F  D3  08             10.0         OUT     RFIFOL           ;CLOCK THE DATA INTO THE FIFO
44A1  3E  F9              7.0         MVI     A,R.PLO0!R.PLO1.R.STPC!$C8  ;LOAD FIFO DATA ONE
                                                                       ;AND SET TIE BUS JAM AND PLO DISABLE
44A3  D3  09             10.0         OUT     RPCTL
44A5  D3  08             10.0         OUT     RFIFOL           ;CLOCK ONES INTO THE FIFO
44A7  05                  4.0         DCR     B                ;DECREMENT THE LOOP COUNT
44A8  C2  9B  44         10.0         JNZ     2$               ;CONTINUE UNTIL ZERO
44AB  3A  87  47         13.0  3$:    LDA     PASCNT           ;LOAD THE PASS # IN CAS
44AE                                   ROUT    R05L
44AE  D3  8A             10.0                 OUT     R05L             ;WRITE AC INTO R05L
```

```
   (1)  44B0  7F                    4.0                 MOV     A,A             ;RETRY LINK
1879    44B1  AF                    4.0         XRA     A               ;PUT THE MACHINE IN PE MODE
1880    44B2  D3  D3               10.0         OUT     WMCCTL
1881    44B4  3E  6A                7.0         MVI     A,FIFORD        ;LOAD THE FIFO READ COMMAND
1882    44B6  D3  0B               10.0         OUT     RCMD
1883    44B8  3E  B8                7.0         MVI     A,R.PL00.R.PL01!R.PLOD!R.TBJN   ;RUN THE RMC
1884                                                                    ;AND SET TIE BUS JAM AND PLO DISABLE
1885    44BA  D3  09               10.0         OUT     RPCTL
1886    44BC  00                    4.0         NOP                     ;WAIT
1887    44BD  00                    4.0         NOP                     ;
1888    44BE  00                    4.0         NOP                     ;
1889    44BF  00                    4.0         NOP                     ;
1890    44C0  3E  B9                7.0         MVI     A,R.PL00!R.PL01!R.STPC!R.PLOD!R.TBJN  ;STOP THE RMC
1891                                                                    ;AND SET TIE BUS JAM AND PLO DISABLE
1892    44C2  D3  09               10.0         OUT     RPCTL
1893    44C4  DB  17               10.0         IN      RDATA
1894    44C6  32  82  47           13.0         STA     DATAA
1895    44C9  DB  15               10.0         IN      RPSTA
1896    44CB  E6  40                7.0         ANI     R.DATA
1897    44CD  07                    4.0         RLC
1898    44CE  07                    4.0         RLC
1899    44CF  32  83  47           13.0         STA     DATAAP
1900    44D2  CD  7F  46           18.0         CALL    CKAMT
1901    44D5  DA  7A  44           10.0         JC      TST03X
1902
1903    44D8  3A  87  47           13.0         LDA     PASCNT
1904    44DB  3C                    4.0         INR     A
1905    44DC  32  87  47           13.0         STA     PASCNT
1906    44DF  3A  84  47           13.0         LDA     DATAE
1907    44E2  EE  FF                7.0         XRI     $FF
1908    44E4  32  84  47           13.0         STA     DATAE
1909    44E7  3A  85  47           13.0         LDA     DATAEP
1910    44EA  EE  01                7.0         XRI     $01
1911    44EC  32  85  47           13.0         STA     DATAEP
1912    44EF  3A  87  47           13.0         LDA     PASCNT
1913    44F2  FE  3E                7.0         CPI     62
1914    44F4  C2  AB  44           10.0         JNZ     3$
1915    44F7                                    ENDTST  TST03X
  (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)   44F7                                    REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
  (2)   44F7  CD  06  28           18.0                                   CALL    REQST
  (2)   44FA  00                                                          .BYTE                   ;DATA PATTERN NUMBER
  (2)   44FB  00  00                                                      .WORD                   ;SYSTEM '*' COUNT
  (2)   44FD  00  00                                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)   44FF  00                                                          .BYTE                   ;DATA COMPARE FLAG IF -1
  (2)   4500  07                                                          .BYTE   7               ;REQUEST CODE
  (1)   4501  3A  9A  4F           13.0                 LDA     ITERA           ;GET ITERATION COUNT
  (1)   4504  3D                    4.0                 DCR     A               ;DOWNCOUNT
  (1)   4505  32  9A  4F           13.0                 STA     ITERA           ;SAVE COUNT
  (1)   4508  F2  7A  44           10.0                 JP      TST03X          ;DO TEST UNTIL TILL = 0
1916    450B  C3  18  28           10.0   EXIT:         JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*************************************************************************
 450E F5                      12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
 450F C5                      12.0          PUSH    B
 4510 06 00                    7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
 4512 DB E0                   10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
 4514 E6 80                    7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
 4516 B0                       4.0          ORA     B               ;ADD IN THE SELECTED PORT #
 4517 D3 E0                   10.0          OUT     MBSEL           ;RESET TO THIS PORT
 4519 3E 80                    7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
 451B D3 40                   10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
 451D AF                       4.0          XRA     A               ;CLEAR TU COMMAND A
 451E D3 40                   10.0          OUT     TCMD            ;
 4520 3E 81                    7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
 4522 D3 40                   10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
 4524 3E 00                    7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
 4526 D3 40                   10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
 4528 AF                       4.0          XRA     A
 4529 D3 44                   10.0          OUT     TAMT            ;CLEAR AMTIE WORD
 452B D3 48                   10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
 452D D3 4C                   10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
 452F 04                       4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
 4530 78                       4.0          MOV     A,B
 4531 FE 04                    7.0          CPI     4               ;DONE?
 4533 C2 12 45                10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
 4536 C1                      10.0          POP     B               ;RESET B & C
 4537 F1                      10.0          POP     PSW             ;ALL DONE
 4538 C9                      10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE CHECK MARK 2
1963  4539                      SSUB
                                ;*************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK READ CHANNELS MARK 2
1965  4539                      SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO CHECK THAT ALL READ CHANNEL 'MARK TWO''
                                ;*BITS EQUAL ONE AND TO FLAG ANY EXCEPTIONS WITH AN ERROR MESSAGE.
1968  4539                      SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*   IF CHANNEL 0=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 0
                                ;*   ENDIF
                                ;*   IF CHANNEL 1=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 1
                                ;*   ENDIF
                                ;*   IF CHANNEL 2=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 2
                                ;*   ENDIF
                                ;*   IF CHANNEL 3-MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 3
                                ;*   ENDIF
                                ;*   IF CHANNEL 4=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 4
                                ;*   ENDIF
                                ;*   IF CHANNEL 5=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 5
                                ;*   ENDIF
                                ;*   IF CHANNEL 6=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 6
                                ;*   FNDIF
                                ;*   IF CHANNEL 7=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL 7
                                ;*   ENDIF
                                ;*   IF CHANNEL P=MARK TWO
                                ;*     THEN-CONTINUE
                                ;*     ELSE-ERROR CHANNEL P
                                ;*   ENDIF
                                ;*ENDSUB
```

```
2007  4539                                    S
 (1)                                          ; ************************************************************************
2008
2009  4539  DB  13          10.0  CKMK:   IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2010  453B  E6  01           7.0          ANI     $01             ;CHANNEL 0 ILLEGAL?
2011  453D  C2  45  45       10.0         JNZ     CKMK0           ;YES-CONTINUE
2012  4540                                 ERR     OUTMK2,CKMK0
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4540  CD  09  28       18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                               MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4543  02                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4544  00                                 .BYTE
 (1)  4545  CD  15  28       18.0                 CKMK0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4548  DA  DB  45       10.0                         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2013                                    ;>CHANNEL 0 NOT MARK TWO
2014
2015  454B  DB  13          10.0         IN      RMK2            ;GET THE READ CHANNEL ILLFGAL BITS
2016  454D  E6  02           7.0          ANI     $02             ;CHANNEL 1 ILLEGAL?
2017  454F  C2  57  45       10.0         JNZ     CKMK1           ;YES-CONTINUE
2018  4552                                 ERR     OUTMK2,CKMK1
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4552  CD  09  28       18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0003                               MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4555  03                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4556  00                                 .BYTE
 (1)  4557  CD  15  28       18.0                 CKMK1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  455A  DA  DB  45       10.0                         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2019                                    ;>CHANNEL 1 NOT MARK TWO
2020
2021  455D  DB  13          10.0         IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2022  455F  E6  04           7.0          ANI     $04             ;CHANNEL 2 ILLEGAL?
2023  4561  C2  69  45       10.0         JNZ     CKMK2           ;YES-CONTINUE
2024  4564                                 ERR     OUTMK2,CKMK2
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4564  CD  09  28       18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                               MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4567  04                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4568  00                                 .BYTE
 (1)  4569  CD  15  28       18.0                 CKMK2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  456C  DA  DB  45       10.0                         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2025                                    ;>CHANNEL 2 NOT MARK TWO
2026  456F  DB  13          10.0         IN      RMK2            ;GET THE READ CHANNFL ILLEGAL BITS
2027  4571  E6  08           7.0          ANI     $08             ;CHANNEL 3 ILLEGAL?
2028  4573  C2  7B  45       10.0         JNZ     CKMK3           ;YES-CONTINUE
2029  4576                                 ERR     OUTMK2,CKMK3
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4576  CD  09  28       18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                               MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4579  05                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  457A  00                                 .BYTE
 (1)  457B  CD  15  28       18.0                 CKMK3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  457E  DA  DB  45       10.0                         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2030                                    ;>CHANNEL 3 NOT MARK TWO
```

```
2032  4581  DB  13       10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2033  4583  E6  10        7.0          ANI     $10             ;CHANNEL 4 ILLEGAL?
2034  4585  C2  8D  45   10.0          JNZ     CKMK4           ;YES-CONTINUE
2035  4588                             ERR     OUTMK2,CKMK4
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4588  CD  09  28   18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  458B  06                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  458C  00                                .BYTE
 (1)  458D  CD  15  28   18.0          CKMK4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4590  DA  DB  45   10.0                  JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2036                    ;>CHANNEL 4 NOT MARK TWO
2037
2038  4593  DB  13       10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2039  4595  E6  20        7.0          ANI     $20             ;CHANNEL 5 ILLEGAL?
2040  4597  C2  9F  45   10.0          JNZ     CKMK5           ;YES-CONTINUE
2041  459A                             ERR     OUTMK2,CKMK5
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  459A  CD  09  28   18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  459D  07                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459E  00                                .BYTE
 (1)  459F  CD  15  28   18.0          CKMK5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45A2  DA  DB  45   10.0                  JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2042                    ;>CHANNEL 5 NOT MARK TWO
2043  45A5  DB  13       10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2044  45A7  E6  40        7.0          ANI     $40             ;CHANNEL 6 ILLEGAL?
2045  45A9  C2  B1  45   10.0          JNZ     CKMK6           ;YES-CONTINUE
2046  45AC                             ERR     OUTMK2,CKMK6
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45AC  CD  09  28   18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45AF  08                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45B0  00                                .BYTE
 (1)  45B1  CD  15  28   18.0          CKMK6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45B4  DA  DB  45   10.0                  JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2047                    ;>CHANNEL 6 NOT MARK TWO
2048
2049  45B7  DB  13       10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2050  45B9  E6  80        7.0          ANI     $80             ;CHANNEL 7 ILLEGAL?
2051  45BB  C2  C3  45   10.0          JNZ     CKMK7           ;YES-CONTINUE
2052  45BE                             ERR     OUTMK2,CKMK7
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45BE  CD  09  28   18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0009                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45C1  09                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45C2  00                                .BYTE
 (1)  45C3  CD  15  28   18.0          CKMK7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45C6  DA  DB  45   10.0                  JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2053                    ;>CHANNEL 7 NOT MARK TWO
```

```
 2055  45C9  DB  15          10.0          IN      RPSTA           ;GET THE CHANNEL P BIT
 2056  45CB  E6  08           7.0          ANI     R.MK2           ;CHANNEL P ILLEGAL?
 2057  45CD  C2  D5  45       10.0          JNZ     CKMKP           ;YES-CONTINUE
 2058  45D0                                ERR     OUTMK2,CKMKP
  (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  45D0  CD  09  28       18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)        000A                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  45D3  0A                             .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  45D4  00                             .BYTE
  (1)  45D5  CD  15  28       18.0          CKMKP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  45D8  DA  DB  45       10.0                  JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
 2059                                 ;>CHANNEL P NOT MARK TWO
 2060
 2061  45DB  C9                10.0  OUTMK2: RET                    ;RETURN
```

```
2063                                .SBTTL  SUBROUTINE CHECK END MARK
2064  45DC                          SSUB
 (1)                                ;*****************************************************************************
 (1)                                ;*SUBROUTINE TITLE
 (1)                                ;*---------- -----
2065                                ;*CHECK READ CHANNELS END MARK
2066  45DC                          SD
 (1)                                ;*****************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
2067                                ;*THIS SUBROUTINE IS USED TO CHECK THAT ALL READ CHANNEL 'END MARK'
2068                                ;*BITS EQUAL ONE AND TO FLAG ANY EXCEPTIONS WITH AN ERROR MESSAGE.
2069  45DC                          SP
 (1)                                ;*****************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
2070                                ;*BGNSUB
2071                                ;*   IF CHANNEL 0=END MARK
2072                                ;*     THEN-CONTINUE
2073                                ;*     ELSE-ERROR CHANNEL 0
2074                                ;*   ENDIF
2075                                ;*   IF CHANNEL 1=END MARK
2076                                ;*     THEN-CONTINUE
2077                                ;*     ELSE-ERROR CHANNEL 1
2078                                ;*   ENDIF
2079                                ;*   IF CHANNEL 2=END MARK
2080                                ;*     THEN-CONTINUE
2081                                ;*     ELSE-ERROR CHANNEL 2
2082                                ;*   ENDIF
2083                                ;*   IF CHANNEL 3=END MARK
2084                                ;*     THEN-CONTINUE
2085                                ;*     ELSE-ERROR CHANNEL 3
2086                                ;*   ENDIF
2087                                ;*   IF CHANNEL 4=END MARK
2088                                ;*     THEN-CONTINUE
2089                                ;*     ELSE-ERROR CHANNEL 4
2090                                ;*   ENDIF
2091                                ;*   IF CHANNEL 5=END MARK
2092                                ;*     THEN-CONTINUE
2093                                ;*     ELSE-ERROR CHANNEL 5
2094                                ;*   ENDIF
2095                                ;*   IF CHANNEL 6 END MARK
2096                                ;*     THEN-CONTINUE
2097                                ;*     ELSE-ERROR CHANNEL 6
2098                                ;*   ENDIF
2099                                ;*   IF CHANNEL 7-END MARK
2100                                ;*     THEN-CONTINUE
2101                                ;*     ELSE-ERROR CHANNEL 7
2102                                ;*   ENDIF
2103                                ;*   IF CHANNEL P=END MARK
2104                                ;*     THEN-CONTINUE
2105                                ;*     ELSE-ERROR CHANNEL P
2106                                ;*   ENDIF
2107                                ;*ENDSUB
```

```
2108  45DC                                  S
 (1)                                        ; ************************************************************************************
2109
2110  45DC   DB   14          10.0  CKEMK:  IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2111  45DE   E6   01           7.0          ANI     $01             ;CHANNEL 0 ILLEGAL?
2112  45E0   C2   E8   45     10.0          JNZ     CKEN0           ;YES-CONTINUE
2113  45E3                                 ERR     OUTEND,CKEN0
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45E3   CD   09   28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)         0008                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45E6   08                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45E7   00                            .BYTE
 (1)  45E8   CD   15   28     18.0  CKEN0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45EB   DA   7E   46     10.0          JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2114                                ;>CHANNEL 0 NOT END MARK
2115
2116  45EE   DB   14          10.0          IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2117  45F0   E6   02           7.0          ANI     $02             ;CHANNEL 1 ILLEGAL?
2118  45F2   C2   FA   45     10.0          JNZ     CKEN1           ;YES-CONTINUE
2119  45F5                                 ERR     OUTEND,CKEN1
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45F5   CD   09   28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)         000C                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45F8   0C                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45F9   00                            .BYTE
 (1)  45FA   CD   15   28     18.0  CKEN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45FD   DA   7E   46     10.0          JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2120                                ;>CHANNEL 1 NOT END MARK
2121
2122  4600   DB   14          10.0          IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2123  4602   E6   04           7.0          ANI     $04             ;CHANNEL 2 ILLEGAL?
2124  4604   C2   0C   46     10.0          JNZ     CKEN2           ;YES-CONTINUE
2125  4607                                 ERR     OUTEND,CKEN2
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4607   CD   09   28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)         000D                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460A   0D                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  460B   00                            .BYTE
 (1)  460C   CD   15   28     18.0  CKEN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  460F   DA   7E   46     10.0          JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2126                                ;>CHANNEL 2 NOT END MARK
2127
2128  4612   DB   14          10.0          IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2129  4614   E6   08           7.0          ANI     $08             ;CHANNEL 3 ILLEGAL?
2130  4616   C2   1E   46     10.0          JNZ     CKEN3           ;YES-CONTINUE
2131  4619                                 ERR     OUTEND,CKEN3
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4619   CD   09   28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)         000E                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  461C   0E                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  461D   00                            .BYTE
 (1)  461E   CD   15   28     18.0  CKEN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4621   DA   7E   46     10.0          JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2132                                ;>CHANNEL 3 NOT END MARK
```

```
2133
2134  4624  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2135  4626  E6  10         7.0        ANI     $10             ;CHANNEL 4 ILLEGAL?
2136  4628  C2  30  46    10.0        JNZ     CKEN4           ;YES-CONTINUE
2137  462B                           ERR     OUTEND,CKEN4
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  462B  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000F                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  462E  0F                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  462F  00                                .BYTE
 (1)  4630  CD  15  28    18.0          CKEN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4633  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2138                            ;>CHANNEL 4 NOT END MARK
2139
2140  4636  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2141  4638  E6  20         7.0        ANI     $20             ;CHANNEL 5 ILLEGAL?
2142  463A  C2  42  46    10.0        JNZ     CKEN5           ;YES-CONTINUE
2143  463D                           ERR     OUTEND,CKEN5
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  463D  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0010                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4640  10                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4641  00                                .BYTE
 (1)  4642  CD  15  28    18.0          CKEN5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4645  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2144                            ;>CHANNEL 5 NOT END MARK
2145
2146  4648  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2147  464A  E6  40         7.0        ANI     $40             ;CHANNEL 6 ILLEGAL?
2148  464C  C2  54  46    10.0        JNZ     CKEN6           ;YES-CONTINUE
2149  464F                           ERR     OUTEND,CKEN6
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  464F  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0011                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4652  11                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4653  00                                .BYTE
 (1)  4654  CD  15  28    18.0          CKEN6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4657  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2150                            ;>CHANNEL 6 NOT END MARK
2151
2152  465A  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2153  465C  E6  80         7.0        ANI     $80             ;CHANNEL 7 ILLEGAL?
2154  465E  C2  66  46    10.0        JNZ     CKEN7           ;YES-CONTINUE
2155  4661                           ERR     OUTEND,CKEN7
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4661  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0012                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4664  12                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4665  00                                .BYTE
 (1)  4666  CD  15  28    18.0          CKEN7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4669  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2156                            ;>CHANNEL 7 NOT END MARK
```

```
2158  466C  DB  15        10.0          IN      RPSTA           ;GET THE CHANNEL P BIT
2159  466E  E6  04         7.0          ANI     R.ILL           ;CHANNEL P ILLEGAL?
2160  4670  C2  78  46     10.0          JNZ     CKENP           ;YES-CONTINUE
2161  4673                               ERR     OUTEND,CKENP
  (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4673  CD  09  28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)       0013                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4676  13                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4677  00                                   .BYTE
  (1)  4678  CD  15  28     18.0          CKENP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  467B  DA  7E  46     10.0                  JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2162                                   ;>CHANNEL P NOT END MARK
2163
2164  467E  C9                 10.0    OUTEND: RET                     ;RETURN
2165
```

```
2167                                    .SBTTL  SUBROUTINE CHECK AMTIE
2168  467F                              SSUB
 (1)                                    ;*****************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
2169                                    ;*CHECK AMTIE SUBROUTINE
2170  467F                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2171                                    ;*THIS SUBROUTINE IS USED TO IS' 'TE A BIT FAILURE IN THE READ PATH DATA
2172                                    ;*OR AMTIE BYTES TO A SPECIFIC RCAD CHANNEL BY COMPARING THE ACTUAL AND
2173                                    ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
2174  467F                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2175                                    ;*BGNSUB
2176                                    ;*     IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
2177                                    ;*       THEN-ERROR CHANNEL 0
2178                                    ;*       ELSE-CONTINUE
2179                                    ;*     ENDIF
2180                                    ;*     IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
2181                                    ;*       THEN-ERROR CHANNEL 1
2182                                    ;*       ELSE-CONTINUE
2183                                    ;*     ENDIF
2184                                    ;*     IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
2185                                    ;*       THEN-ERROR CHANNEL 2
2186                                    ;*       ELSE-CONTINUE
2187                                    ;*     ENDIF
2188                                    ;*     IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
2189                                    ;*       THEN-ERROR CHANNEL 3
2190                                    ;*       ELSE-CONTINUE
2191                                    ;*     ENDIF
2192                                    ;*     IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
2193                                    ;*       THEN-ERROR CHANNEL 4
2194                                    ;*       ELSE-CONTINUE
2195                                    ;*     ENDIF
2196                                    ;*     IF CHANNEL 5 ACTUAL INFORMATION NOT - CHANNEL 5 EXPECTED
2197                                    ;*       THEN-ERROR CHANNEL 5
2198                                    ;*       ELSE-CONTINUE
2199                                    ;*     ENDIF
2200                                    ;*     IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
2201                                    ;*       THEN-ERROR CHANNEL 6
2202                                    ;*       ELSE-CONTINUE
2203                                    ;*     ENDIF
2204                                    ;*     IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
2205                                    ;*       THEN-ERROR CHANNEL 7
2206                                    ;*       ELSE-CONTINUE
2207                                    ;*     ENDIF
2208                                    ;*     IF CHANNEL P ACTUAL INFORMATION NOT - CHANNEL P EXPECTED
2209                                    ;*       THEN-ERROR CHANNEL P
2210                                    ;*       ELSE-CONTINUE
2211                                    ;*     ENDIF
```

```
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************
2212                                    CKAMT:  LXI     H,DATAE         ;LOAD EXPECTED DATA ADDRESS
2213  467F                              
 (1)                                    
2214  467F  21  84  47        10.0             LDA     DATAA           ;GET THE ACTUAL DATA
2215  4682  3A  82  47        13.0             ROUT    ADATA           ;WRITE THE ACTUAL DATA TO THE CAS
2216  4685                                    
 (1)  4685  D3  94             10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4687  7F                  4.0                      MOV     A,A             ;RETRY LINK
2217  4688  E6  01              7.0             ANI     $01             ;REMOVE UNWANTED BITS
2218  468A  47                  4.0             MOV     B,A             ;SAVE IN B
2219  468B  7E                  7.0             MOV     A,M             ;GET EXPECTED DATA
2220  468C                                    ROUT    EDATA           ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  468C  D3  95             10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  468E  7F                  4.0                      MOV     A,A             ;RETRY LINK
2221  468F  E6  01              7.0             ANI     $01             ;REMOVE UNWANTED BITS
2222  4691  B8                  4.0             CMP     B               ;COMPARE
2223  4692  CA  9A  46          10.0             JZ      CKAM0           ;CONTINUE IF EQUAL
2224  4695                                    ERRB    OUT,CKAM0,1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4695  CD  12  28        18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0014                                      MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4698  14                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4699  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  469A  CD  15  28        18.0             CKAM0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  469D  DA  63  47        10.0                     JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2225                                    ;>CHANNEL 0 FAILED
2226                                    
2227  46A0  3A  82  47        13.0             LDA     DATAA           ;GET THE ACTUAL DATA
2228  46A3  E6  02              7.0             ANI     $02             ;REMOVE UNWANTED BITS
2229  46A5  47                  4.0             MOV     B,A             ;SAVE IN B
2230  46A6  7E                  7.0             MOV     A,M             ;GET EXPECTED DATA
2231  46A7  E6  02              7.0             ANI     $02             ;REMOVE UNWANTED BITS
2232  46A9  B8                  4.0             CMP     B               ;COMPARE
2233  46AA  CA  B2  46          10.0             JZ      CKAM1           ;CONTINUE IF EQUAL
2234  46AD                                    ERRB    OUT,CKAM1,1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46AD  CD  12  28        18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0015                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4680  15                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4681  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4682  CD  15  28        18.0             CKAM1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4685  DA  63  47        10.0                     JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2235                                    ;>CHANNEL 1 FAILED
```

```
 2237  46B8  3A  82  47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
 2238  46BB  E6  04           7.0          ANI     $04             ;REMOVE UNWANTED BITS
 2239  46BD  47               4.0          MOV     B,A             ;SAVE IN B
 2240  46BE  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
 2241  46BF  E6  04           7.0          ANI     $04             ;REMOVE UNWANTED BITS
 2242  46C1  B8               4.0          CMP     B               ;COMPARE
 2243  46C2  CA  CA  46      10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
 2244  46C5                                ERRB    OUT,CKAM2,1
  (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  46C5  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0016                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46C8  16                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46C9  01                                      .BYTE   1               ;PRINT ROUTINE NUMBER
  (1)  46CA  CD  15  28      18.0          CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46CD  DA  63  47      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
 2245                            ;>CHANNEL 2 FAILED
 2246  46D0  3A  82  47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
 2247  46D3  E6  08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
 2248  46D5  47               4.0          MOV     B,A             ;SAVE IN B
 2249  46D6  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
 2250  46D7  E6  08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
 2251  46D9  B8               4.0          CMP     B               ;COMPARE
 2252  46DA  CA  E2  46      10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
 2253  46DD                                ERRB    OUT,CKAM3,1
  (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  46DD  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0017                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46E0  17                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46E1  01                                      .BYTE   1               ;PRINT ROUTINE NUMBER
  (1)  46E2  CD  15  28      18.0          CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46E5  DA  63  47      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
 2254                            ;>CHANNEL 3 FAILED
 2255
 2256  46E8  3A  82  47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
 2257  46EB  E6  10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
 2258  46ED  47               4.0          MOV     B,A             ;SAVE IN B
 2259  46EE  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
 2260  46EF  E6  10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
 2261  46F1  B8               4.0          CMP     B               ;COMPARE
 2262  46F2  CA  FA  46      10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
 2263  46F5                                ERRB    OUT,CKAM4,1
  (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1)  46F5  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0018                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46F8  18                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46F9  01                                      .BYTE   1               ;PRINT ROUTINE NUMBER
  (1)  46FA  CD  15  28      18.0          CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46FD  DA  63  47      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
 2264                            ;>CHANNEL 4 FAILED
```

```
2266  4700  3A  82  47  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2267  4703  E6  20      7.0          ANI     $20             ;REMOVE UNWANTED BITS
2268  4705  47          4.0          MOV     B,A             ;SAVE IN B
2269  4706  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2270  4707  E6  20      7.0          ANI     $20             ;REMOVE UNWANTED BITS
2271  4709  B8          4.0          CMP     B               ;COMPARE
2272  470A  CA  12  47  10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
2273  470D                           ERRB    OUT,CKAM5,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  470D  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0019                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4710  19                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4711  01                                 .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4712  CD  15  28  18.0            CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4715  DA  63  47  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2274                            ;>CHANNEL 5 FAILED
2275
2276  4718  3A  82  47  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2277  471B  E6  40      7.0          ANI     $40             ;REMOVE UNWANTED BITS
2278  471D  47          4.0          MOV     B,A             ;SAVE IN B
2279  471E  7E          7 0          MOV     A,M             ;GET EXPECTED DATA
2280  471F  E6  40      7.C          ANI     $40             ;REMOVE UNWANTED BITS
2281  4721  B8          4.0          CMP     B               ;COMPARE
2282  4722  CA  2A  47  10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
2283  4725                           ERRB    OUT,CKAM6,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4725  CD  12  28  '8.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001A                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4728  1A                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4729  01                                 .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  472A  CD  15  28  18.0            CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  472D  DA  63  47  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2284                            ;>CHANNEL 6 FAILED
2285
2286  4730  3A  82  47  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2287  4733  E6  80      7.0          ANI     $80             ;REMOVE UNWANTED BITS
2288  4735  47          4.0          MOV     B,A             ;SAVE IN B
2289  4736  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2290  4737  E6  80      7.0          ANI     $80             ;REMOVE UNWANTED BITS
2291  4739  B8          4.0          CMP     B               ;COMPARE
2292  473A  CA  42  47  10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
2293  473D                           ERRB    OUT,CKAM7,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  473D  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001B                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4740  1B                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4741  01                                 .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4742  CD  15  28  18.0            CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4745  DA  63  47  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2294                            ;>CHANNEL 7 FAILED
```

```
 2296  4748  21  85  47      10.0          LXI     H,DATAEP        ;GET THE EXPECTED DATA PARITY ADDRESS
 2297  474B  3A  83  47      13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
 2298  474E  E6  01           7.0          ANI     $01             ;REMOVE UNWANTED BITS
 2299  4750  47               4.0          MOV     B,A             ;SAVE IN B
 2300  4751  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
 2301  4752  E6  01           7.0          ANI     $0              ;REMOVE UNWANTED BITS
 2302  4754  B8               4.0          CMP     B               ;COMPARE
 2303  4755  CA  5D  47      10.0          JZ      CKAMP           ;CONTINUE IF EQUAL
 2304  4758                                ERRB    OUT,CKAMP,1
   (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
   (1)  4758  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
   (1)       001C                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
   (1)  475B  1C                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
   (1)  475C  01                                    .BYTE   1               ;PRINT ROUTINE NUMBER
   (1)  475D  CD  15  28      18.0          CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
   (1)  4760  DA  63  47      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
 2305                                  ;>CHANNEL P FAILED
 2306
 2307  4763  C9               10.0  OUT:     RET
```

```
2309                                    .SBTTL  TABLE INPUT DATA MARK TWO
2310
2311  4764  FF          M8950MK: .BYTE  @377        ;DATA GROUP 1 - CHARACTER 1
2312  4765  01                   .BYTE  @001        ;DATA GROUP 1 - CHARACTER 1  PARITY
2313  4766  FF                   .BYTE  @377        ;DATA GROUP 1 - CHARACTER 2
2314  4767  01                   .BYTE  @001        ;DATA GROUP 1 - CHARACTER 2  PARITY
2315  4768  FF                   .BYTE  @377        ;DATA GROUP 1 - CHARACTER 3
2316  4769  01                   .BYTE  @001        ;DATA GROUP 1 - CHARACTER 3  PARITY
2317  476A  00                   .BYTE  @000        ;DATA GROUP 1 - CHARACTER 4
2318  476B  00                   .BYTE  @000        ;DATA GROUP 1 - CHARACTER 4  PARITY
2319  476C  00                   .BYTE  @000        ;DATA GROUP 1 - CHARACTER 5
2320  476D  80                   .BYTE  @200        ;DATA GROUP 1 - CHARACTER 5  PARITY
2321
2322                                    .SBTTL  TABLE INPUT DATA END MARK
2323  476E  FF          M8950EM: .BYTE  @377        ;DATA GROUP 1 - CHARACTER 1
2324  476F  01                   .BYTE  @001        ;DATA GROUP 1 - CHARACTER 1  PARITY
2325  4770  FF                   .BYTE  @377        ;DATA GROUP 1 - CHARACTER 2
2326  4771  01                   .BYTE  @001        ;DATA GROUP 1 - CHARACTER 2  PARITY
2327  4772  FF                   .BYTE  @377        ;DATA GROUP 1 - CHARACTER 3
2328  4773  01                   .BYTE  @001        ;DATA GROUP 1 - CHARACTER 3  PARITY
2329  4774  FF                   .BYTE  @377        ;DATA GROUP 1 - CHARACTER 4
2330  4775  01                   .BYTE  @001        ;DATA GROUP 1 - CHARACTER 4  PARITY
2331  4776  FF                   .BYTE  @377        ;DATA GROUP 1 - CHARACTER 5
2332  4777  81                   .BYTE  @201        ;DATA GROUP 1 - CHARACTER 5  PARITY
```

```
2334                                    .SBTTL  PROGRAM VARIABLES
2335  4778   00                         UNITMP: .BYTE   0               ;UNIT MAP
2336  4779   00                         SUNIT:  .BYTE   0               ;PORT/TU SELECT BYTE
2337  477A   00    00                   AMTMSK: .WORD   0               ;AMTIE MASK WORD
2338  477C   00    00                   INTBLTMP: .WORD 0               ;TEMP. STORAGE FOR THE INPUT TABLE POINTER
2339  477E   00    00                   INTBL:  .WORD   0               ;INPUT TABLE ADDRESS STORAGE
2340  4780   00    00                   OUTTBL: .WORD   0               ;OUTPUT TABLE ADDRESS STORAGE
2341  4782   00                         DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2342  4783   00                         DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2343  4784   00                         DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2344  4785   00                         DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2345  4786   00                         SUBCNT: .BYTE   0               ;SUBGROUP COUNTER
2346  4787   00                         PASCNT: .BYTE   0               ;PASS COUNTER
2347         0000                               .END
```

RPM5.M80 SYMBOL TABLE

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK  477A           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B         CBYTL  = 008A          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093         CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D         CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKAMP   475D  G        CKAMT   467F          CKAM0   469A  G       CKAM1   4682  G
CKAM2   46CA  G        CKAM3   46E2  G       CKAM4   46FA  G       CKAM5   4712  G
CKAM6   472A  G        CKAM7   4742  G       CKEMK   45DC          CKENP   4678  G
CKEN0   45E8  G        CKEN1   45FA  G       CKEN2   460C  G       CKEN3   461E  G
CKEN4   4630  G        CKEN5   4642  G       CKEN6   4654  G       CKEN7   4666  G
CKLOP  = 2815          CKMK    4539          CKMKP   45D5  G       CKMK0   4545  G
CKMK1   4557  G        CKMK2   4569  G       CKMK3   457B  G       CKMK4   458D  G
CKMK5   459F  G        CKMK6   45B1  G       CKMK7   45C3  G       CLEAR   450E
CLKCTL- 00F0           CLOCK   4F26          CLRLP   4512          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B         CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F         CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090         CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E         CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082         C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010         C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070         C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040         C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          D      =%0002         DATAA   4782          DATAAP  4783
DATACT= 00D0           DATAE   4784          DATAEP  4785          DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 000B         DONF1  = 0045          DONINT= 0010
DSAVE   4F9E           DSE    = 0006         DUMMY   431E  G       D.ATH0= 0001
D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020          D.NOTW= 0040
D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080          E      =%0003
ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041          ECCSTA= 001A
ECCTST= 000F           EDATA  = 0095         EOTCLR= 0003          ERFLG   4F93
ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812          ERLPE  = 280C
ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F          EXIT    4508
E.ACRC= 0010           E.AMT  = 0020         E.CDP  = 0080          E.CRC  = 0080
E.PNTR- 0008           E.RPE  = 0040         E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25          FOUND   4344
FWDTST- 0061           GCRID  = 0089         GCRSET= 0002          GOODTM= 0092
H      =%0004          HLSAVE  4FA0          IE     = 0008          INTBL   477E
INTBLT  477C           INTSTA- 00E0          ITERA   4F9A          I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010         I6.5   = 0020          I7.5   = 0040
KCALL  = 005F          KCLR   = 007B         KDEP   = 003F          KENAB  = 0078
```

```
KEXAM = 003E            KEYBRD= 00C8            KEY1  = 0078            KEY10 = 006D
KEY11 = 006E            KEY12 = 006F            KEY13 = 005C            KEY14 = 005D
KEY15 = 005E            KEY16 = 005F            KEY17 = 003C            KEY18 = 003D
KEY19 = 003E            KEY2  = 0079            KEY20 = 003F            KEY3  = 007A
KEY4  = 007B            KEY5  = 0074            KEY6  = 0075            KEY7  = 0076
KEY8  = 0077            KEY9  = 006C            KINTA = 006F            KLDAD = 003D
KN0   = 003C            KN1   = 005C            KN2   = 005D            KN3   = 005E
KN4   = 006C            KN5   = 006D            KN6   = 006E            KN7   = 0074
KN8   = 0075            KN9   = 0076            KU2   = 0079            KU3   = 007A
KU8   = 0077            L     =%0005            LBLANK= 000F            LCE   = 0008
LCH   = 000C            LCL   = 000D            LCP   = 000E            LC0   = 0000
LC1   = 0001            LC2   = 0002            LC3   = 0003            LC4   = 0004
LC5   = 0005            LC6   = 0006            LC7   = 0007            LC8   = 0008
LC9   = 0009            LDLEDA= 00CA            LDLEDB= 00CB            LDLEDC= 00CC
LDLEDD= 00CD            LDLEDE= 00CE            LDLEDF= 00CF            LKDIAG= 2800
LKKBD = 004C            LKKEY = 0049            LKLWMG= 0058            LKLWMP= 0055
LKLWPG= 0052            LKLWPP= 004F            LKMOD7= 0046            LKOPR = 0046
LPFLG   4F94            LPNUM   4F92            M     =%0006            MBSEL = 00E0
MB.A  = 0008            MB.B  = 0004            MEMTOP= 4FFF            MINUS = 000A
MM    = 8000            MSE   = 0008            MSGN  = 001C            MTACLR= 0000
MT.ARA= 0020            MT.CPE= 0080            MT.DSE= 0001            MT.FWD= 0040
MT.INH= 0008            MT.LWR= 0004            MT.MOT= 0002            MT.NWT= 0080
MT.PEC= 0040            MT.PSB= 0004            MT.PS0= 0001            MT.PS1= 0002
MT.REV= 0020            MT.WRT= 0010            MT.Z  = 0008            M.ATA = 0080
M.CAPE= 0020            M.CONT= 0080            M.DEM = 0020            M.EBL = 0010
M.EXC = 0008            M.FAIL= 0008            M.ILR = 0010            M.INIT= 0010
M.OCC = 0020            M.ONLI= 0001            M.PE  = 0040            M.PORT= 0080
M.RDEN= 0002            M.RDPE= 0008            M.RUN = 0004            M.SCLK= 0001
M.TRA = 0040            M.UNIT= 0007            M.WCLK= 0040            M.WCLN= 0080
M.WREN= 0080            M5.5  = 0001            M6.5  = 0002            M7.5  = 0004
M8950E  476E            M8950M  4764            NOTCAP= 0088            OKAY  = 00FF
OPRRAM= 4300            OPSTRT= 0058            OPVER = 0040            OUT     4763
OUTEND  467E            OUTMK2  45DB            OUTTBL  4780            PADCNT= 00D5
PADCRC= 0080            PASCNT  4787            PDIAG = 0048            PEID  = 008A
PENAB = 004C            PESET = 0001            PL    = 00B1            PRDD  = 004C
PRENF = 009C            PS    = 00B2            PSTAT = 0048            PSW   =%0009
P.AMTP= 0001            P.BCTC= 0040            P.CMDP= 0020            P.INTE= 0080
P.LCS = 0040            P.LWR = 0020            P.RDP = 0002            P.RPEN= 0004
P.RPST= 0002            P.RP0E= 0020            P.RP1E= 0010            P.RP2E= 0020
P.RP3E  0010            P.SING= 0080            P.STAT= 0002            P.STPE= 0080
P.TACH= 0008            P.TUPR= 0010            P.WCSP= 0004            P.WDS = 0040
P.WFLP- 0001            P.WPEN= 0010            P.WP0E= 0008            P.WP1E= 0004
P.WP2E= 0008            P.WP3E= 0004            P.5VOK= 0002            QUE   = 281B
QUEM  = 281E            RAMT  = 0010            RARA  = 0006            RARAI = 0004
RCHBD0= 0048            RCHBD1= 0047            RCHOK = 0046            RCHTST= 000C
RCLRT = 000D            RCMD  = 000B            RCMLP = 0003            RCONT = 0080
RDATA = 0017            RDCLK = 0010            RDON  = 0011            READG = 0007
REND  = 0014            REQST = 2806            RESCHR= 00D1            REVTST= 0064
REWIND- 0004            RFIFOL= 0008            RGCLK = 0002            RGCRI = 0003
RIBG  = 0001            RILL  = 0012            RINST = 000C            RMCTST= 0008
RMK2  = 0013            RNOP  = 0000            RPATH = 0001            RPBAD = 0044
RPCHI = 0001            RPCLK = 0003            RPCTL = 0009            RPEI  = 0002
RPFAIL= 0000            RPF1  = 009D            RPF2  = 009E            RPOK  = 0043
RPOSTN= 0016            RPSTA = 0015            RRCMT = 000A            RSTAT = 0002
```

```
RTIEB = 000A            RTIER = 0030            RTM    = 0005           RUNKI = 0009
RUPTST= 005E            RWDUNL= 0005            R.AMT  = 0001           R.BOP = 0008
R.DATA= 0040            R.DON = 0002            R.DRDY= 0010            R.END = 0010
R.ILL = 0004            R.JVOK= 0004            R.MK2  = 0008           R.PLOD= 0008
R.PLO0= 0010            R.PLO1= 0020            R.POST= 0020            R.STNM= 0002
R.STOP= 0004            R.STPC= 0001            R.TBJN= 0080            R.TSTD= 0040
R.VOK = 0080            R00H  = 0081            R00L   = 0080           R01H  = 0083
R01L  = 0082            R02H  = 0085            R02L   = 0084           R03H  = 0087
R03L  = 0086            R04H  = 0089            R04L   = 0088           R05H  = 008B
R05L  = 008A            R06H  = 008D            R06L   = 008C           R07H  = 008F
R07L  = 008E            R10H  = 0091            R10L   = 0090           R11H  = 0093
R11L  = 0092            R12H  = 0095            R12L   = 0094           R13H  = 0097
R13L  = 0096            R14H  = 0099            R14L   = 0098           R15H  = 009B
R15L  = 009A            R16H  = 009D            R16L   = 009C           R17H  = 009F
R17L  = 009E            R7.5  = 0010            SELCLR= 0000            SETATA= 00A1
SID   = 0080            SOD   = 0080            SOE    = 0040           SP    =%0008
SSCLK = 0040            SSTEP = 0005            STACK  = 4FFF           STATRM= 4F20
STPCT   4F20            STRSP = 5000            SUBCNT   4786           SUNIT   4779
TADR00= 0080            TADR01= 0081            TADR02= 0082            TADR03= 0083
TADR04= 0084            TADR05= 0085            TADR06= 0086            TADR07= 0087
TADR10= 0088            TADR11= 0089            TADR12= 008A            TADR13= 008B
TAMT  = 0044            TASEL = 0080            TCMD   = 0040           TC.FWD= 0040
TC.INH= 0008            TC.LWR= 0004            TC.REV= 0020            TC.WRT= 0010
TEMP    4F99            TEST1   4300            TEST2    43DE           TEST3   4475
TMF   = 0099            TMRDY = 0040            TRKENA= 00D2            TSET  = 2803
TSTEND= 2818            TSTS  = 0040            TS101A   43BF           TST01C  4377
TST01X  4351            TST02A  4456            TST02C   440E           TST02X  43E8
TST03X  447A            TUSEL0= 00D1            TUSEL1= 00D2            TU78  = 0010
T.ATH0= 0001            T.ATH1= 0002            T.BOT  = 0004           T.EOT = 0002
T.FPT = 0001            T.NTHR= 0004            T.ONL  = 0020           T.PES = 0008
T.PSBJ= 0020            T.PS0J= 0008            T.PS1J= 0010            T.RDY = 0080
T.RDYO= 0040            T.RWD = 0010            T.SCLK= 0002            UIBG  = 00A1
UNITMP  4778            UNITSL  4327            VALFC    4F98           VALTB   4F95
VELTST= 005B            WDR.P = 0010            WRTCLK= 0000            WMCERR= 00DA
WMCSTA= 00D0            WRTCLK= 0000            WRTDAT= 00D3            W.ACRC= 0004
W.CRC = 0008            W.DIAG= 0002            W.DONN= 0040            W.ECC = 0010
W.ENAB= 0080            W.ERR = 0020            W.FMT  = 0070           W.GCR = 0010
W.LEFT= 0004            W.ONES= 0020            W.RESI= 0002            W.REV = 0004
W.ROME= 0010            W.RST = 0001            W.SKIP= 000F            W.WRIT= 0008
W.XFER- 0020            X     =%000A            X.DONN= 0080            X.ENAB= 0040
X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001            Y     =%0008
.     - 4788

ERRORS DETECTED:  0

*RPM5.A78/PTP,RPM5=NLIST,PARAM,MACRO,LIST,RPM5
RUN-TIME: 4 6 0 SECONDS
CORE USED:  10K
```

```
                                .TITLE  ECC1 - ECC CONTROLLER PART #1
                                .SBTTL  TEST 1 - ECC NO CORRECTION TEST
                                :ID     ECC1-ERROR CORRECTION CONTROLLER PART #1
 4300                           ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ECC NO CORRECTION TEST
 4300                           SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PASS DATA THROUGH
                                ;*WITHOUT PERFORMING ANY CORRECTION ON THE DATA.  THE DATA INPUT IS CORRECT
                                ;*IN EVERY DETAIL I.E. 5 TO 4 TRANSLATION, PROPER ECC CHARACTER, AND NO
                                ;*TRACK IN ERROR POINTERS.
 4300                           SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*    CLEAR THE TIE BUS
                                ;*    CLEAR THE INPUT DATA
                                ;*  BGNDO
                                ;*  : ENABLE THE READ PATH CLOCK
                                ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  : CLOCK THE FIFO'S
                                ;*  : ISSUE CLEAR ALL COMMAND
                                ;*  : WAIT
                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*  : CALL SUBROUTINE CLECC
                                ;*  : SET THE LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  GET THE INPUT DATA BYTE
                                ;*  : :  CALL SUBROUTINE ECC
                                ;*  : :  DECREMENT THE LOOP COUNT
                                ;*  : :  DO UNTIL THE LOOP COUNT=0
                                ;*  : ENDDO
                                ;*  : SET THE LOOP COUNT TO 3
                                ;*  : BGNDO
                                ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
                                ;*  : :  CALL SUBROUTINE T4X5
                                ;*  : :  INIT THE LOOP COUNT TO 5
                                ;*  : :  BGNDO
                                ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                                ;*  : :  :  DECREMENT THE LOOP COUNT
```

```
1374                    ;*  : : :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1375                    ;*  : : :  DO UNTIL LOOP COUNT = 0
1376                    ;*  : : ENDDO
1377                    ;*  : : FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
1378                    ;*  : : FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1379                    ;*  : : CALL SUBROUTINE T4X5
1380                    ;*  : : INIT THE LOOP COUNT TO 5
1381                    ;*  : : BGNDO
1382                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1383                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1384                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1385                    ;*  : : :  DECREMENT THE LOOP COUNT
1386                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1387                    ;*  : : ENDDO
1388                    ;*  : : DECREMENT THE LOOP COUNT
1389                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1390                    ;*  : ENDDO
1391                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1392                    ;*  : SET UP WATCHDOG TIMER COUNT
1393                    ;*  : BGNDO
1394                    ;*  : :    SINGLE STEP THE READ PATH
1395                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1396                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1397                    ;*  : ENDDO
1398                    ;*  : IF WATCHDOG TIMER=0
1399                    ;*  : :    THEN-ERROR
1400                    ;*  : :    ELSE-CONTINUE
1401                    ;*  : ENDIF
1402                    ;*  : SINGLE STEP THE READ PATH
1403                    ;*  : CALL SUBROUTINE CKDONE
1404                    ;*  : INIT THE LOOP COUNT TO 8
1405                    ;*  : BGNDO
1406                    ;*  : :    SINGLE STEP THE READ PATH
1407                    ;*  : :    INPUT THE ACTUAL DATA
1408                    ;*  : :    INPUT THE ACTUAL PARITY
1409                    ;*  : :    CALL SUBROUTINE CK5X4
1410                    ;*  : :    DECREMENT THE LOOP COUNT
1411                    ;*  : :    DO UNTIL LOOP COUNT = 0
1412                    ;*  : ENDDO
1413                    ;*  : INPUT THE ECC CHARACTER FROM THE READ CHANNELS
1414                    ;*  : IF ACTUAL ECC CHARACTER=EXPECTED ECC CHARACTER
1415                    ;*  : :  THEN-CONTINUE
1416                    ;*  : :  ELSE-ERROR
1417                    ;*  : ENDIF
1418                    ;*  : SET UP WATCHDOG TIMER COUNT
1419                    ;*  : BGNDO
1420                    ;*  : :  SINGLE STEP THE READ PATH
1421                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1422                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1423                    ;*  : ENDDO
1424                    ;*  : IF WATCHDOG TIMER=0
1425                    ;*  : :  THEN-ERROR-EXIT TEST
1426                    ;*  : :  ELSE-CONTINUE
1427                    ;*  : ENDIF
```

```
                                ;*   : SINGLE STEP THE READ PATH
                                ;*   : INIT THE  LOOP COUNT TO 7
                                ;*   : BGNDO
                                ;*   : :  SINGLE STEP THE READ PATH
                                ;*   : :  COMPARE CORRECTED DATA WITH INPUT DATA
                                ;*   : :  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
                                ;*   : .  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  DECREMENT THE LOOP COUNT
                                ;*   : :  DO UNTIL THE LOOP COUNT = 0
                                ;*   : ENDDO
                                ;*   : SINGLE STEP THE READ PATH
                                ;*   : IF EXPECTED NOT EQUAL
                                ;*   : :  THEN-ECC ERROR
                                ;*   : :  ELSE-CONTINUE
                                ;*   : ENDIF
                                ;*   : INPUT THE ECC STATUS
                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*   : IF THE RESULTING STATUS=0
                                ;*   : :  THEN-CONTINUE
                                ;*   : :  ELSE-ERROR
                                ;*   : ENDIF
                                ;*   : INCREMENT THE INPUT DATA
                                ;*   : DO UNTIL INPUT DATA - 0.
                                ;*  ENDDO
                                ;*ENDTST
1460  4300                      SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1461                            ;*ECC1 MICRO TEST 01
1462                            ;*ECC1 MICRO ERROR 01
1463                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1464                            ;*M8951, M8950'S
1465                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1466                            ;*FATAL ERROR - TEST ABORTED
1467                            ;*
1468                            ;*ECC1 MICRO TEST 01
1469                            ;*ECC1 MICRO ERROR 02
1470                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1471                            ;*M8951, M8950'S
1472                            ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1473                            ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1474                            ;*BYTE/SCLK COUNT NUMBER = LLL
1475                            ;*FATAL ERROR - MICRO TEST ABORTED
1476                            ;*
1477                            ;*ECC1 MICRO TEST 01
1478                            ;*ECC1 MICRO ERROR 03
```

```
1479                                    ;*ECC1-CONTROLLER-NO CORRECTION TEST
1480                                    ;*M8951, M8950'S
1481                                    ;*ECC CHARACTER INCORRECT AT OUTPUT OF M8950'S
1482                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1483                                    ;*ACTUAL = NNNN
1484                                    ;*EXPECTED = NNNN
1485                                    ;*
1486                                    ;*ECC1 MICRO TEST 01
1487                                    ;*ECC1 MICRO ERROR 04
1488                                    ;*ECC1-CONTROLLER-NO CORRECTION TEST
1489                                    ;*M8951, M8950'S
1490                                    ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1491                                    ;*SENT TO THE ECC
1492                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1493                                    ;*FATAL ERROR - MICRO TEST ABORTED
1494                                    ;*
1495                                    ;*ECC1 MICRO TEST 01
1496                                    ;*ECC1 MICRO ERROR 05
1497                                    ;*ECC1-CONTROLLER-NO CORRECTION TEST
1498                                    ;*M8951, M8950'S
1499                                    ;*CORRECTED DATA FROM THE ECC-INCORRECT
1500                                    ;*NO CORRECTION SHOULD HAVE TAKEN PLACE
1501                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1502                                    ;*ACTUAL = NNNN
1503                                    ;*EXPECTED = NNNN
1504                                    ;*
1505                                    ;*ECC1 MICRO TEST 01
1506                                    ;*ECC1 MICRO ERROR 06
1507                                    ;*ECC1-CONTROLLER-NO CORRECTION TEST
1508                                    ;*M8951, M8950'S
1509                                    ;*CORRECTED DATA PARITY BIT = ZERO SHOULD   ONE
1510                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1511                                    ;*
1512                                    ;*ECC1 MICRO TEST 01
1513                                    ;*ECC1 MICRO ERROR 07
1514                                    ;*ECC1-CONTROLLER-NO CORRECTION TEST
1515                                    ;*M8951, M8950'S
1516                                    ;*CORRECTED DATA PARITY BIT = ONE SHOULD - ZERO
1517                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1518                                    ;*
1519                                    ;*ECC1 MICRO TEST 01
1520                                    ;*ECC1 MICRO ERROR 10
1521                                    ;*ECC1-CONTROLLER-NO CORRECTION TEST
1522                                    ;*M8951, M8950'S
1523                                    ;*CORRECTED ECC CHARACTER INCORRECT
1524                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1525                                    ;*
1526                                    ;*ECC1 MICRO TEST 01
1527                                    ;*ECC1 MICRO ERROR 11
1528                                    ;*ECC1-CONTROLLER-NO CORRECTION TEST
1529                                    ;*M8951, M8950'S
1530                                    ;*ECC STATUS INCORRECT AFTER A DATA GROUP
1531                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1532                                    ;*ACTUAL = NNNN
```

```
1533                            ;*EXPECTED = NNNN
1534                            ;*
1535                            ;*ECC1 MICRO TEST 01
1536                            ;*ECC1 MICRO ERROR 22
1537                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1538                            ;*M8951, M8950'S
1539                            ;*CHANNEL 0 FAILED
1540                            ;*BYTE/SCLK COUNT NUMBER = LLL
1541                            ;*ACTUAL = NNNN
1542                            ;*EXPECTED = NNNN
1543                            ;*
1544                            ;*ECC1 MICRO TEST 01
1545                            ;*ECC1 MICRO ERROR 23
1546                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1547                            ;*M8951, M8950'S
1548                            ;*CHANNEL 1 FAILED
1549                            ;*BYTE/SCLK COUNT NUMBER = LLL
1550                            ;*ACTUAL = NNNN
1551                            ;*EXPECTED = NNNN
1552                            ;*
1553                            ;*ECC1 MICRO TEST 01
1554                            ;*ECC1 MICRO ERROR 24
1555                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1556                            ;*M8951, M8950'S
1557                            ;*CHANNEL 2 FAILED
1558                            ;*BYTE/SCLK COUNT NUMBER = LLL
1559                            ;*ACTUAL = NNNN
1560                            ;*EXPECTED = NNNN
1561                            ;*
1562                            ;*ECC1 MICRO TEST 01
1563                            ;*ECC1 MICRO ERROR 25
1564                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1565                            ;*M8951, M8950'S
1566                            ;*CHANNEL 3 FAILED
1567                            ;*BYTE/SCLK COUNT NUMBER = LLL
1568                            ;*ACTUAL = NNNN
1569                            ;*EXPECTED = NNNN
1570                            ;*
1571                            ;*ECC1 MICRO TEST 01
1572                            ;*ECC1 MICRO ERROR 26
1573                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1574                            ;*M8951, M8950'S
1575                            ;*CHANNEL 4 FAILED
1576                            ;*BYTE/SCLK COUNT NUMBER = LLL
1577                            ;*ACTUAL = NNNN
1578                            ;*EXPECTED = NNNN
1579                            ;*
1580                            ;*ECC1 MICRO TEST 01
1581                            ;*ECC1 MICRO ERROR 27
1582                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1583                            ;*M8951, M8950'S
1584                            ;*CHANNEL 5 FAILED
1585                            ;*BYTE/SCLK COUNT NUMBER   LLL
1586                            ;*ACTUAL = NNNN
```

```
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC1 MICRO TEST 01
                                        ;*ECC1 MICRO ERROR 30
                                        ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
                                        ;*M8951, M8950'S
                                        ;*CHANNEL 6 FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC1 MICRO TEST 01
                                        ;*ECC1 MICRO ERROR 31
                                        ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
                                        ;*M8951, M8950'S
                                        ;*CHANNEL 7 FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC1 MICRO TEST 01
                                        ;*ECC1 MICRO ERROR 32
                                        ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
                                        ;*M8951, M8950'S
                                        ;*CHANNEL P FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
 (1)                                    ; ************************************************************************************
1616
1617 4300                               TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1) 4300  3E  01            7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1618                                    ;%ECC1-ECC CONTROLLER-NO CORRECTION TEST
1619                                    ;&M8951, M8950'S, M8952
1620 4305                                       REQ     @7,0,0,0,0
 (1) 4305  CD  06  28       18.0                 CALL    REQST
 (1) 4308  00                                    .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4309  00  00                                .WORD   0               ;SYSTEM "0" COUNT
 (1) 430B  00  00                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D  00                                    .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1) 430E  07                                    .BYTE   @7              ;REQUEST CODE
1621 430F                                       RIN     R12L
 (1) 430F  DB  94           10.0                 IN      R12L            ;READ R12L INTO AC
 (1) 4311  7F                4.0                 MOV     A,A             ;RETRY LINK
1622 4312  32  26  4A       13.0         STA     UNITMP
1623 4315  A7                4.0         ANA     A
1624 4316  C2  27  43       10.0         JNZ     UNITSL
```

```
                                                ERR     EXIT,DUMMY
                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
                                                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
                                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                        .BYTE
                                                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
                                                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;>FATAL ERROR - TEST ABORTED
                                        JMP     EXIT

                                UNITSL: MVI     B,@0
                                        LDA     UNITMP          ;DID THE USER SPECIFY TU
                                        ANI     @01             ;PORT 0?
                                        JNZ     FOUND           ;YES-GO USE PORT #0
                                        INR     B               ;NO-UPDATE POINTER TO PORT #1
                                        LDA     UNITMP          ;DID THE USER SPECIFY TU
                                        ANI     @02             ;PORT 1?
                                        JNZ     FOUND           ;YES-GO USE PORT #1
                                        INR     B               ;NO-UPDATE POINTER TO PORT #2
                                        LDA     UNITMP          ;DID THE USER SPECIFY TU
                                        ANI     @04             ;PORT 2?
                                        JNZ     FOUND           ;YES-GO USE PORT #2
                                        INR     B               ;NO-ASSUME PORT #3
                                FOUND:  CALI    CLEAR
                                        IN      INTSTA
                                        ANI     BIT7
                                        ORA     B
                                        OUT     MBSEL
                                        STA     SUNIT
                                        MVI     A,P.RPEN
                                        OUT     PENAB

                                TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
                                        OUT     CLKCTL

                                        MVI     A,W.GCR         ;SET GCR MODE
                                        OUT     WMCCTL

                                        MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
                                        OUT     PDIAG           ;LCS, LWR AND DPEN
                                        XRA     A
                                        OUT     TAMT
                                        OUT     TRKENA          ;CLEAR ALL TRACKS FROM
                                        OUT     TRKENA          ;THE TRANSLATOR
                                        STA     INDAT
                                        ROUT    R05H
                                                OUT     R05H            ;WRITE AC INTO R05H
                                                MOV     A,A             ;RETRY LINK

                                TST01C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
                                                                        ;SET TIE BUS JAM AND PLO DISABLE
```

```
1671  4370  D3  09        10.0          OUT   RPCTL
1672  4372  AF             4.0          XRA   A
1673  4373  D3  0A        10.0          OUT   RTIEB          ;CLEAR THE TIE BUS
1674  4375  D3  08        10.0          OUT   RFIFOL         ;CLOCK THE FIFO'S
1675  4377  3E  0D         7.0          MVI   A,RCLRT        ;ISSUE CLEAR ALL COMMAND
1676  4379  D3  0B        10.0          OUT   RCMD
1677
1678  437B  00             4.0          NOP                  ;WAIT
1679  437C  00             4.0          NOP                  ;WAIT
1680  437D  00             4.0          NOP
1681  437E  00             4.0          NOP
1682  437F  00             4.0          NOP
1683
1684  4380  3E  A9         7.0          MVI   A,R.PLO1!R.STPC!R.PLOD.R.TBJN   ;STOP THE READ PATH
1685                                                                          ;SET TIE BUS JAM AND PLO DISABLE
1686  4382  D3  09        10.0          OUT   RPCTL
1687  4384  CD  50  48    18.0          CALL  CLECC          ;CLEAR THE ECC CHARACTER
1688  4387  16  07         7.0          MVI   D,@7           ;SET UP THE LOOP COUNT
1689  4389  3A  24  4A    13.0  TST01F: LDA   INDAT          ;GET THE INPUT DATA
1690  438C  CD  55  48    18.0          CALL  ECC            ;CALCULATE ECC FOR
1691  438F  15             4.0          DCR   D              ;THE 7 INPUT BYTES
1692  4390  C2  89  43    10.0          JNZ   TST01F
1693
1694  4393  0E  03         7.0          MVI   C,@3           ;INIT THE LOOP COUNT
1695  4395  3A  24  4A    13.0  TST01I: LDA   INDAT          ;GET THE INPUT DATA
1696  4398                              ROUT  R05L
 (1)  4398  D3  8A        10.0                OUT   R05L               ;WRITE AC INTO R05L
 (1)  439A  7F             4.0                MOV   A,A                ;RETRY LINK
1697  439B  32  40  48    13.0          STA   TRNIN          ;FILL THE TRANSLATOR
1698  439E  32  41  48    13.0          STA   TRNIN+1        ;SUBROUTINE INPUT
1699  43A1  32  42  48    13.0          STA   TRNIN+2        ;BUFFER
1700  43A4  32  43  48    13.0          STA   TRNIN+3        ;
1701  43A7  CD  C0  47    18.0          CALL  T4X5           ;TRANSLATE THE SUBGROUP
1702
1703  43AA  21  44  48    10.0          LXI   H,TRNOUT       ;GET POINTER TO TRANSLATE DATA TABLE
1704  43AD  06  05         7.0          MVI   B,@5           ;SET UP LOOP COUNT
1705  43AF  7E             7.0  TST01G: MOV   A,M            ;GET A DATA BYTE
1706  43B0  D3  40        10.0          OUT   TCMD           ;STORE DATA IN COMMAND ADDRESS
1707  43B2  23             6.0          INX   H              ;POINT TO DATA PARITY
1708  43B3  7E             7.0          MOV   A,M            ;GET THE DATA PARITY
1709
1710  43B4  07             4.0          RLC                  ;POSITION FOR OUTPUT
1711  43B5  F6  60         7.0          ORI   P.LWR.P.LCS    ;OR IN CONTROL BITS
1712  43B7  D3  48        10.0          OUT   PDIAG          ;OUTPUT THE DATA PARITY
1713
1714  43B9  D3  08        10.0          OUT   RFIFOL         ;CLOCK DATA INTO THE FIFO'S
1715
1716  43BB  23             6.0          INX   H              ;UPDATE THE TABLE POINTER
1717  43BC  05             4.0          DCR   B              ;DECREMENT LOOP COUNT
1718  43BD  C2  AF  43    10.0          JNZ   TST01G         ;DO UNTIL LOOP COUNT = 0
1719
1720  43C0  3A  24  4A    13.0          LDA   INDAT          ;GET THE INPUT DATA
1721  43C3  32  40  48    13.0          STA   TRNIN          ;FILL THE FIRST THREE
1722  43C6  32  41  48    13.0          STA   TRNIN+1        ;BYTES OF THE TRANSLATOR
```

```
1723  43C9  32  42  48      13.0          STA   TRNIN+2          ;SUBROUTINE WITH THE INPUT DATA
1724  43CC  3A  78  48      13.0          LDA   ECCCHR           ;STORE THE CALCULATED
1725  43CF  32  43  48      13.0          STA   TRNIN+3          ;ECC CHARACTER AS THE LAST CHARACTER
1726
1727  43D2  CD  C0  47      18.0          CALL  T4X5             ;TRANSLATE THE SECOND SUBGROUP
1728
1729  43D5  21  44  48      10.0          LXI   H,TRNOUT         ;GET THE POINTER TO INPUT DATA TABLE
1730  43D8  06  05           7.0          MVI   B,@5             ;SET UP THE LOOP COUNT
1731  43DA  7E               7.0  TST01H: MOV   A,M             ;GET A DATA BYTE
1732  43DB  D3  40          10.0          OUT   TCMD             ;STORE DATA IN CMD ADDRESS
1733  43DD  23               6.0          INX   H                ;POINT TO DATA PARITY BIT
1734  43DE  7E               7.0          MOV   A,M              ;GET THE DATA PARITY
1735  43DF  07               4.0          RLC                    ;POSITION FOR OUTPUT
1736  43E0  F6  60           7.0          ORI   P.LWR!P.LCS      ;OR IN CONTROL BITS
1737  43E2  D3  48          10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
1738  43E4  D3  08          10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFOS
1739  43E6  23               6.0          INX   H                ;UPDATE THE TABLE POINTER
1740  43F7  05               4.0          DCR   B                ;DECREMENT THE LOOP COUNT
1741  43E8  C2  DA  43       10.0          JNZ   TST01H           ;DO UNTIL LOOP COUNT=0
1742
1743  43EB  0D               4.0          DCR   C                ;PUT THE GROUP IN
1744  43EC  C2  95  43       10.0          JNZ   TST01I           ;THE FIFO TWICE
1745
1746  43EF  3E  0B           7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
1747  43F1  D3  0B          10.0          OUT   RCMD             ;COMMAND
1748  43F3  11  01  00      10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
1749  43F6  21  A8  FD      10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
1750  43F9  D3  0C          10.0  2$:     OUT   RINST            ;STEP THE READ PATH
1751  43FB  DB  01          10.0          IN    RPCHI            ;DATA READY SET?
1752  43FD  E6  10           7.0          ANI   R.DRDY
1753  43FF  C2  14  44       10.0          JNZ   TST01A           ;YES-GO PROCESS
1754  4402  19               10.0          DAD   D                ;WATCH DOG TIMEOUT?
1755  4403  D2  F9  43       10.0          JNC   2$               ;NO-CONTINUE
1756  4406                                ERR   TST01C,DUMMX,1
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4406  CD  09  28      18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0002                            MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4409  02                              .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  440A  01                              .BYTE 1
 (1)  440B  CD  15  28      18.0    DUMMX:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  440E  DA  6E  43      10.0            JC    TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1757                              ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1758                              ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1759                              ;<FATAL ERROR - MICRO TEST ABORTED
1760  4411  C3  92  47      10.0          JMP   EXIT
1761
1762  4414  0E  08           7.0  TST01A: MVI   C,$8             ;INIT THE LOOP COUNT
1763  4416  00               4.0  TST01B: NOP
1764  4417  DB  17          10.0          IN    RDATA            ;INPUT THE ACTUAL DATA BYTE
1765  4419  32  2A  4A       13.0          STA   DATAA            ;STORE IT
1766  441C  DB  15          10.0          IN    RPSTA            ;INPUT THE ACTUAL PARITY BIT
1767  441E  E6  40           7.0          ANI   R.DATA           ;REMOVE JUNK BITS
1768  4420  07               4.0          RLC                    ;POSITION THE BIT FOR COMPARE
1769  4421  07               4.0          RLC
```

```
1770  4422  32 2B 4A   13.0          STA   DATAAP          ;STORE IT
1771  4425  3A 24 4A   13.0          LDA   INDAT           ;GET THE EXPECTED DATA BYTE
1772  4428  32 2C 4A   13.0          STA   DATAE           ;STORE IT
1773  442B  A7          4.0          ANA   A               ;SET CONDITION BITS
1774  442C  E2 34 44   10.0          JPO   1$              ;ODD # OF ONES - CLEAR PARITY
1775  442F  3E 01        7.0          MVI   A,@1            ;EVEN # OF ONES - SET PARITY
1776  4431  C3 35 44   10.0          JMP   2$
1777  4434  AF          4.0    1$:   XRA   A
1778  4435  32 2D 4A   13.0    2$:   STA   DATAEP
1779  4438  CD 8B 48   18.0          CALL  CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
1780  443B  DA 6E 43   10.0          JC    TST01C          ;IF LOOP ON ERROR REPEAT THE TEST
1781  443E  0D          4.0          DCR   C               ;DECREMENT THE LOOP COUNT
1782  443F  D3 0C       10.0          OUT   RINST           ;SINGLE STEP THE READ PATH
1783  4441  C2 16 44   10.0          JNZ   TST01B          ;DO UNTIL LOOP COUNT = ZERO
1784  4444  DB 17       10.0          IN    RDATA           ;INPUT THE ACTUAL ECC CHARACTER
1785  4446                        ROUT  ADATA           ;STORE IN CAS
 (1)  4446  D3 94       10.0                OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  4448  7F          4.0                 MOV   A,A             ;RETRY LINK
1786  4449  47          4.0          MOV   B,A             ;SAVE IN B
1787  444A  3A 78 48   13.0          LDA   ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1788  444D                        ROUT  EDATA           ;STORE IN CAS
 (1)  444D  D3 95       10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  444F  7F          4.0                 MOV   A,A             ;RETRY LINK
1789  4450  B8          4.0          CMP   B               ;ACTUAL=EXPECTED?
1790  4451  CA 59 44   10.0          JZ    ECCCK1          ;YES-CONTINUE
1791  4454                        ERRB  TST01C,ECCCK1,1
 (1)                            ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4454  CD 12 28   18.0                CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)  0003                         MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4457  03                     .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4458  01                     .BYTE 1               ;PRINT ROUTINE NUMBER
 (1)  4459  CD 15 28   18.0    ECCCK1::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  445C  DA 6E 43   10.0                JC    TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1792                            ;>ECC CHARACTER INCORRECT AT OUTPUT OF M8950'S
1793                            ;<
1794  445F  11 01 00   10.0          LXI   D,1             ;SET WATCHDOG TIMER INCREMENT
1795  4462  21 A8 FD   10.0          LXI   H,-600          ;SET WATCHDOG COUNT TO 600
1796  4465  D3 0C       10.0    3$:   OUT   RINST           ;SINGLE STEP THE READ PATH
1797  4467  DB 01       10.0          IN    RPCHI           ;DATA READY SET?
1798  4469  E6 10        7.0          ANI   R.DRDY
1799  446B  C2 80 44   10.0          JNZ   TST01K          ;YES-GO PROCESS
1800  446E  19          10.0          DAD   D               ;NO-WATCHDOG TIMEOUT?
1801  446F  D2 65 44   10.0          JNC   3$              ;NO-CONTINUE
1802  4472                        ERR   TST01C,DUMMW,1
 (1)                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4472  CD 09 28   18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)  0004                         MSGN  -     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4475  04                     .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4476  01                     .BYTE 1
 (1)  4477  CD 15 28   18.0    DUMMW:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  447A  DA 6E 43   10.0                JC    TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1803                            ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1804                            ;>SENT TO THE ECC
1805                            ;<FATAL ERROR - MICRO TEST ABORTED
```

```
 1806  447D  C3  92  47      10.0          JMP     EXIT
 1807
 1808  4480  16  01          7.0  TST01K:  MVI     D,@1            ;INIT THE LOOP COUNT
 1809  4482  AF              4.0  TST01Y:  XRA     A
 1810  4483                          ROUT    R05H
  (1)  4483  D3  8B          10.0                 OUT     R05H            ;WRITE AC INTO R05H
  (1)  4485  7F              4.0                  MOV     A,A             ;RETRY LINK
 1811  4486  3A  24  4A      13.0          LDA     INDAT           ;GET THE INPUT DATA
 1812  4489                          ROUT    EDATA
  (1)  4489  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  448B  7F              4.0                  MOV     A,A             ;RETRY LINK
 1813  448C  A7              4.0          ANA     A               ;SET CONDITION BITS
 1814  448D  3E  00          7.0          MVI     A,0             ;CLEAR THE ACC.
 1815  448F  E2  93  44      10.0         JPO     1$              ;ODD # OF ONES - CONTINUE
 1816  4492  3C              4.0          INR     A               ;EVEN # OF ONES - SET PARITY
 1817  4493  32  25  4A      13.0  1$:     STA     INDATP          ;STORE THE PARITY BIT
 1818  4496  D3  0C          10.0  TST01L:  OUT     RINST           ;SINGLE STEP THE READ MACHINE
 1819  4498  3A  24  4A      13.0          LDA     INDAT
 1820  449B  47              4.0          MOV     B,A
 1821  449C  DB  19          10.0          IN      ECCCOR
 1822  449E  B8              4.0          CMP     B
 1823  449F  CA  AE  44      10.0          JZ      TST01M
 1824  44A2                          ROUT    ADATA
  (1)  44A2  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  44A4  7F              4.0                  MOV     A,A             ;RETRY LINK
 1825  44A5  78              4.0          MOV     A,B             ;GET THE BYTE COUNT
 1826  44A6                          ROUT    R05L            ;STORE IN CAS
  (1)  44A6  D3  8A          10.0                 OUT     R05L            ;WRITE AC INTO R05L
  (1)  44A8  7F              4.0                  MOV     A,A             ;RETRY LINK
 1827  44A9                          ERRB    TST01C,TST01M,@1
  (1)                             ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
  (1)  44A9  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        0005                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  44AC  05                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  44AD  01                                 .BYTE   @1              ;PRINT ROUTINE NUMBER
  (1)  44AE  CD  15  28      18.0         TST01M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  44B1  DA  6E  43      10.0                 JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
 1828                         ;>CORRECTED DATA FROM THE ECC-INCORRECT
 1829                         ;>NO CORRECTION SHOULD HAVE TAKEN PLACE
 1830  44B4  DB  15          10.0          IN      RPSTA           ;GET THE ECC PARITY BIT
 1831  44B6  E6  80          7.0           ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
 1832  44B8  07              4.0           RLC                     ;POSITION BIT FOR COMPARE
 1833  44B9  4F              4.0           MOV     C,A             ;SAVE IN REG. C
 1834  44BA  3A  25  4A      13.0          LDA     INDATP          ;GET THE EXPECTED PARITY BIT
 1835  44BD  A7              4.0           ANA     A               ;SET THE CONDITION BITS
 1836  44BE  CA  D9  44      10.0          JZ      TST01QO         ;GO CHECK FOR PARITY=0
 1837  44C1  79              4.0           MOV     A,C             ;ELSE CHECK FOR PARITY=1
 1838  44C2  FE  01          7.0           CPI     @1
 1839  44C4  CA  D0  44      10.0          JZ      TST01RG
 1840  44C7  78              4.0           MOV     A,B
 1841  44C8                          ROUT    R05L
  (1)  44C8  D3  8A          10.0                 OUT     R05L            ;WRITE AC INTO R05L
  (1)  44CA  7F              4.0                  MOV     A,A             ;RETRY LINK
```

```
                                        ERR     TST01C,TST01RG,@1
                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
                                                CALL    ERLP            ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE   @1
                                        TST01RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
                                                JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
                                ;<

                                        JMP     TST01SZ         ;CONTINUE WITH TEST
                                TST01QO: MOV    A,C             ;CHECK FOR PARITY = ZERO
                                        ANA     A               ;SET CONDITION BITS
                                        JZ      TST01TP         ;CONTINUE IF ZERO
                                        MOV     A,B
                                        ROUT    R05L
                                                OUT     R05L            ;WRITE AC INTO R05L
                                                MOV     A,A             ;RETRY LINK
                                        ERR     TST01C,TST01TP,@1
                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
                                                CALL    ERLP            ;PROCESS ERROR - DO 2.3
                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
                                                .BYTE   @1
                                        TST01TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
                                                JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                ;<

                                TST01SZ: MOV    A,D             ;GET THE CHARACTER COUNT
                                        INR     A               ;INCREMENT
                                        CPI     $8              ;DONE?
                                        JZ      TST01N          ;YES-GO CHECK ECC
                                        MOV     D,A             ;NO-CONTINUE
                                        NOP
                                        NOP
                                        JMP     TST01Y

                                TST01N: OUT     RINST           ;SINGLE STEP THE READ PATH
                                        LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
                                        ROUT    EDATA           ;SAVE IN CAS
                                                OUT     EDATA           ;WRITE AC INTO EDATA
                                                MOV     A,A             ;RETRY LINK
                                        MOV     B,A             ;SAVE IN REGISTER B
                                        IN      ECCCOR
                                        ROUT    ADATA
                                                OUT     ADATA           ;WRITE AC INTO ADATA
                                                MOV     A,A             ;RETRY LINK
                                        CMP     B
                                        JZ      TST01U
                                        LDA     INDAT           ;GET THE DATA/COUNT
                                        ROUT    R05L
                                                OUT     R05L            ;WRITE AC INTO R05L
```

```
  (1)  4511  7F                  4.0                     MOV     A,A             ;RETRY LINK
1876   4512  AF                  4.0             XRA     A
1877   4513                                      ROUT    R05H
  (1)  4513  D3  8B              10.0                     OUT     R05H            ;WRITE AC INTO R05H
  (1)  4515  7F                   4.0                     MOV     A,A             ;RETRY LINK
1878   4516                                      ERR     TST01C,TST01U,1
  (1)                                             ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4516  CD  09  28          18.0                             CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)        0008                                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4519  08                                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  451A  01                                                   .BYTE   1
  (1)  451B  CD  15  28          18.0                     TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  451E  DA  6E  43          10.0                             JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1879                                        ;>CORRECTED ECC CHARACTER INCORRECT
1880                                        ;<
1881
1882   4521  DB  1A                10.0             IN      ECCSTA          ;GET THE ECC STATUS
1883   4523  E6  0F                 7.0             ANI     @17             ;REMOVE UNWANTED BITS
1884   4525  CA  3A  45            10.0             JZ      TST01V
1885   4528                                      ROUT    ADATA
  (1)  4528  D3  94                10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  452A  7F                    4.0                     MOV     A,A             ;RETRY LINK
1886   452B  AF                    4.0             XRA     A
1887   452C                                      ROUT    EDATA
  (1)  452C  D3  95                10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  452E  7F                    4.0                     MOV     A,A             ;RETRY LINK
1888   452F  3A  24  4A            13.0             LDA     INDAT           ;GET THE DATA/COUNT
1889   4532                                      ROUT    R05L            ;SAVE IT
  (1)  4532  D3  8A                10.0                     OUT     R05L            ;WRITE AC INTO R05L
  (1)  4534  7F                    4.0                     MOV     A,A             ;RETRY LINK
1890   4535                                      ERRB    TST01C,TST01V,1
  (1)                                             ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
  (1)  4535  CD  12  28          18.0                             CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        0009                                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4538  09                                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4539  01                                                   .BYTE   1               ;PRINT ROUTINE NUMBER
  (1)  453A  CD  15  28          18.0                     TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  453D  DA  6E  43          10.0                             JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1891                                        ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1892                                        ;<
1893
1894   4540  3A  24  4A            13.0             LDA     INDAT
1895   4543  3C                    4.0             INR     A
1896   4544  32  24  4A            13.0             STA     INDAT
1897   4547  C2  6E  43            10.0             JNZ     TST01C
1898   454A                                      ENDTST  TST01X
  (1)                                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  454A                                   REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  454A  CD  06  28          18.0                             CALL    REQST
  (2)  454D  00                                                   .BYTE                   ;DATA PATTERN NUMBER
  (2)  454E  00  00                                               .WORD                   ;SYSTEM '=' COUNT
  (2)  4550  00  00                                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4552  00                                                   .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4553  07                                                   .BYTE   7               ;REQUEST CODE
```

```
   (1)  4554  3A  9A  4F    13.0    LDA   ITERA    ;GET ITERATION COUNT
   (1)  4557  3D              4.0    DCR   A        ;DOWNCOUNT
   (1)  4558  32  9A  4F    13.0    STA   ITERA    ;SAVE COUNT
   (1)  455B  F2  55  43    10.0    JP    TST01X   ;DO TEST UNTIL TILL = 0
1899    455E  CD  95  47    18.0    CALL  CLEAR    ;CLEAR ALL TU PORTS
```

```
1901                                    .SBTTL  TEST 2 - ECC SOFT POINTERS NO BAD DATA
1902  4561                              ST
 (1)                                    ;*********************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1903                                    ;*ECC SOFT POINTERS NO BAD DATA
1904  4561                              SD
 (1)                                    ;*********************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1905                                    ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PASS GOOD DATA THROUGH
1906                                    ;*WITHOUT PERFORMING ANY CORRECTION ON THE DATA, IN SPITE OF THE
1907                                    ;*PRESENCE OF SOFT TIE POINTERS.  THE DATA INPUT IS CORRECT
1908                                    ;*IN EVERY DETAIL I.E. 5 TO 4 TRANSLATION, PROPER ECC CHARACTER, AND
1909                                    ;*DIFFERENT TRACK IN ERROR POINTERS.
1910  4561                              SP
 (1)                                    ;*********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1911                                    ;*BGNTST
1912                                    ;*    SET NORMAL READ PATH CLOCK
1913                                    ;*    CALL SUBROUTINE CLEAR
1914                                    ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1915                                    ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1916                                    ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1917                                    ;*    CLEAR THE INPUT DATA
1918                                    ;*  BGNDO
1919                                    ;*  : ENABLE THE READ PATH CLOCK
1920                                    ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1921                                    ;*  : CLOCK THE FIFO'S
1922                                    ;*  : ISSUE CLEAR ALL COMMAND
1923                                    ;*  : WAIT
1924                                    ;*  : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1925                                    ;*  : CALL SUBROUTINE CLECC
1926                                    ;*  : SET THE LOOP COUNT TO 7
1927                                    ;*  : BGNDO
1928                                    ;*  : :  GET THE INPUT DATA BYTE
1929                                    ;*  : :  CALL SUBROUTINE ECC
1930                                    ;*  : :  DECREMENT THE LOOP COUNT
1931                                    ;*  : :  DO UNTIL THE LOOP COUNT=0
1932                                    ;*  : ENDDO
1933                                    ;*  : SET THE LOOP COUNT TO 2
1934                                    ;*  : BGNDO
1935                                    ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
1936                                    ;*  : :  CALL SUBROUTINE T4X5
1937                                    ;*  : :  INIT THE LOOP COUNT TO 5
1938                                    ;*  : :  BGNDO
1939                                    ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1940                                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1941                                    ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1942                                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1943                                    ;*  : :  :  DECREMENT THE LOOP COUNT
1944                                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1945                                    ;*  : :  :  DO UNTIL LOOP COUNT = 0
```

```
1946                                    ;*  : :  ENDDO
1947                                    ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
1948                                    ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1949                                    ;*  : :  CALL SUBROUTINE T4X5
1950                                    ;*  : :  INIT THE LOOP COUNT TO 5
1951                                    ;*  : :  BGNDO
1952                                    ;*  : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1953                                    ;*  : :  :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1954                                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1955                                    ;*  : :  :  DECREMENT THE LOOP COUNT
1956                                    ;*  : :  :  DO UNTIL THE LOOP COUNT = 0
1957                                    ;*  : :  ENDDO
1958                                    ;*  : :  DECREMENT THE LOOP COUNT
1959                                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
1960                                    ;*  : ENDDO
1961                                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1962                                    ;*  : SET UP WATCHDOG TIMER COUNT
1963                                    ;*  : BGNDO
1964                                    ;*  : :    SINGLE STEP THE READ PATH
1965                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1966                                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1967                                    ;*  : ENDDO
1968                                    ;*  : IF WATCHDOG TIMER=0
1969                                    ;*  : :    THEN-ERROR
1970                                    ;*  : :    ELSE-CONTINUE
1971                                    ;*  : ENDIF
1972                                    ;*  : GET THE TIE BUS TABLE ADDRESS
1973                                    ;*  : SINGLE STEP THE READ PATH
1974                                    ;*  : INIT THE LOOP COUNT TO 8
1975                                    ;*  : BGNDO
1976                                    ;*  : :    LOAD THE TIE BUS REGISTER FROM THE TABLE
1977                                    ;*  : :    INCREMENT THE TIE BUS DATA TABLE POINTER
1978                                    ;*  : :    SINGLE STEP THE READ PATH
1979                                    ;*  : :    INPUT THE ACTUAL DATA
1980                                    ;*  : :    INPUT THE ACTUAL PARITY
1981                                    ;*  : :    CALL SUBROUTINE CK5X4
1982                                    ;*  : :    DECREMENT THE LOOP COUNT
1983                                    ;*  : :    DO UNTIL LOOP COUNT = 0
1984                                    ;*  : ENDDO
1985                                    ;*  : LOAD THE TIE BUS REGISTER FROM THE TABLE
1986                                    ;*  : INPUT THE ECC CHARACTER FROM THE READ CHANNELS
1987                                    ;*  : IF ACTUAL ECC CHARACTER=EXPECTED ECC CHARACTER
1988                                    ;*  : :    THEN-CONTINUE
1989                                    ;*  : :    ELSE-ERROR
1990                                    ;*  : ENDIF
1991                                    ;*  : SET UP WATCHDOG TIMER COUNT
1992                                    ;*  : BGNDO
1993                                    ;*  : :  SINGLE STEP THE READ PATH
1994                                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1995                                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1996                                    ;*  : ENDDO
1997                                    ;*  : IF WATCHDOG TIMER=0
1998                                    ;*  : :  THEN-ERROR-EXIT TEST
1999                                    ;*  : :  ELSE-CONTINUE
```

```
2000                                ;*   : ENDIF
2001                                ;*   : SINGLE STEP THE READ PATH
2002                                ;*   : CALL SUBROUTINE CKDONE
2003                                ;*   : INIT THE  LOOP COUNT TO 7
2004                                ;*   : BGNDO
2005                                ;*   : :  SINGLE STEP THE READ PATH
2006                                ;*   : :  COMPARE CORRECTED DATA WITH INPUT DATA
2007                                ;*   : :  IF NOT EQUAL
2008                                ;*   : :  :  THEN-ECC ERROR
2009                                ;*   : :  :  ELSE-CONTINUE
2010                                ;*   : :  ENDIF
2011                                ;*   : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
2012                                ;*   : :  IF NOT EQUAL
2013                                ;*   : :  :  THEN-ECC ERROR
2014                                ;*   : :  :  ELSE-CONTINUE
2015                                ;*   : :  ENDIF
2016                                ;*   : :  DECREMENT THE LOOP COUNT
2017                                ;*   : :  DO UNTIL THE LOOP COUNT = 0
2018                                ;*   : ENDDO
2019                                ;*   : SINGLE STEP THE READ PATH
2020                                ;*   : IF EXPECTED NOT EQUAL
2021                                ;*   : :  THEN-ECC ERROR
2022                                ;*   : :  ELSE-CONTINUE
2023                                ;*   : ENDIF
2024                                ;*   : INPUT THE ECC STATUS
2025                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
2026                                ;*   : IF THE RESULTING STATUS=0
2027                                ;*   : :  THEN-CONTINUE
2028                                ;*   : :  ELSE-ERROR
2029                                ;*   : ENDIF
2030                                ;*   : INCREMENT THE INPUT DATA
2031                                ;*   : DO UNTIL INPUT DATA = 0
2032                                ;*  ENDDO
2033                                ;*ENDTST
2034  4561                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
2035                                ;*ECC1 MICRO TEST 02
2036                                ;*ECC1 MICRO ERROR 12
2037                                ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2038                                ;*M8951, M8950'S
2039                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
2040                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2041                                ;*BYTE/SCLK COUNT NUMBER = LLL
2042                                ;*FATAL ERROR - MICRO TEST ABORTED
2043                                ;*
2044                                ;*ECC1 MICRO TEST 02
2045                                ;*ECC1 MICRO ERROR 13
2046                                ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2047                                ;*M8951, M8950'S
2048                                ;*ECC CHARACTER INCORRECT AT OUTPUT OF M8950'S
2049                                ;*BYTE/SCLK COUNT NUMBER = LLL
2050                                ;*ACTUAL = NNNN
```

```
                        ;*EXPECTED = NNNN
                        ;*
                        ;*ECC1 MICRO TEST 02
                        ;*ECC1 MICRO ERROR 14
                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                        ;*M8951, M8950'S
                        ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                        ;*SENT TO THE ECC
                        ;*BYTE/SCLK COUNT NUMBER = LLL
                        ;*FATAL ERROR - MICRO TEST ABORTED
                        ;*
                        ;*ECC1 MICRO TEST 02
                        ;*ECC1 MICRO ERROR 15
                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                        ;*M8951, M8950'S
                        ;*CORRECTED DATA FROM THE ECC-INCORRECT
                        ;*NO CORRECTION SHOULD HAVE TAKEN PLACE
                        ;*BYTE/SCLK COUNT NUMBER = LLL
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*ECC1 MICRO TEST 02
                        ;*ECC1 MICRO ERROR 16
                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                        ;*M8951, M8950'S
                        ;*CORRECTED DATA PARITY BIT - ZERO SHOULD = ONE
                        ;*BYTE/SCLK COUNT NUMBER = LLL
                        ;*
                        ;*ECC1 MICRO TEST 02
                        ;*ECC1 MICRO ERROR 17
                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                        ;*M8951, M8950'S
                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                        ;*BYTE/SCLK COUNT NUMBER = LLL
                        ;*
                        ;*ECC1 MICRO TEST 02
                        ;*ECC1 MICRO ERROR 20
                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                        ;*M8951, M8950'S
                        ;*CORRECT ECC CHARACTER INCORRECT
                        ;*BYTE/SCLK COUNT NUMBER = LLL
                        ;*
                        ;*ECC1 MICRO TEST 02
                        ;*ECC1 MICRO ERROR 21
                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                        ;*M8951, M8950'S
                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                        ;*BYTE/SCLK COUNT NUMBER = LLL
                        ;*ACTUAL = NNNN
                        ;*EXPECTED = NNNN
                        ;*
                        ;*ECC1 MICRO TEST 02
                        ;*ECC1 MICRO ERROR 22
                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
```

```
2105                            ;*M8951, M8950'S
2106                            ;*CHANNEL 0 FAILED
2107                            ;*BYTE/SCLK COUNT NUMBER = LLL
2108                            ;*ACTUAL = NNNN
2109                            ;*EXPECTED = NNNN
2110                            ;*
2111                            ;*ECC1 MICRO TEST 02
2112                            ;*ECC1 MICRO ERROR 23
2113                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2114                            ;*M8951, M8950'S
2115                            ;*CHANNEL 1 FAILED
2116                            ;*BYTE/SCLK COUNT NUMBER = LLL
2117                            ;*ACTUAL = NNNN
2118                            ;*EXPECTED = NNNN
2119                            ;*
2120                            ;*ECC1 MICRO TEST 02
2121                            ;*ECC1 MICRO ERROR 24
2122                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2123                            ;*M8951, M8950'S
2124                            ;*CHANNEL 2 FAILED
2125                            ;*BYTE/SCLK COUNT NUMEER = LLL
2126                            ;*ACTUAL = NNNN
2127                            ;*EXPECTED = NNNN
2128                            ;*
2129                            ;*ECC1 MICRO TEST 02
2130                            ;*ECC1 MICRO ERROR 25
2131                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2132                            ;*M8951, M8950'S
2133                            ;*CHANNEL 3 FAILED
2134                            ;*BYTE/SCLK COUNT NUMBER = LLL
2135                            ;*ACTUAL = NNNN
2136                            ;*EXPECTED = NNNN
2137                            ;*
2138                            ;*ECC1 MICRO TEST 02
2139                            ;*ECC1 MICRO ERROR 26
2140                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2141                            ;*M8951, M8950'S
2142                            ;*CHANNEL 4 FAILED
2143                            ;*BYTE/SCLK COUNT NUMBER - LLL
2144                            ;*ACTUAL = NNNN
2145                            ;*EXPECTED = NNNN
2146                            ;*
2147                            ;*ECC1 MICRO TEST 02
2148                            ;*ECC1 MICRO ERROR 27
2149                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2150                            ;*M8951, M8950'S
2151                            ;*CHANNEL 5 FAILED
2152                            ;*BYTE/SCLK COUNT NUMBER = LLL
2153                            ;*ACTUAL = NNNN
2154                            ;*EXPECTED = NNNN
2155                            ;*
2156                            ;*ECC1 MICRO TEST 02
2157                            ;*ECC1 MICRO ERROR 30
2158                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
```

```
                                        ;*M8951, M8950'S
                                        ;*CHANNEL 6 FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC1 MICRO TEST 02
                                        ;*ECC1 MICRO ERROR 31
                                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                        ;*M8951, M8950'S
                                        ;*CHANNEL 7 FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC1 MICRO TEST 02
                                        ;*ECC1 MICRO ERROR 32
                                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                        ;*M8951, M8950'S
                                        ;*CHANNEL P FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4561                                    S
 (1)                                    ; ***************************************************************************
                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1) 4561  3E  02            7.0                 MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1) 4563  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                        ;&M8951, M8950'S

                                        TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
4566  3E  10                7.0                OUT     CLKCTL
                                        ;SET GCR MODE
456A  3E  10                7.0                MVI     A,W.GCR
456C  D3  D3               10.0                OUT     WMCCTL
456E  CD  95  47           18.0                CALL    CLEAR           ;CLEAR ALL TU PORTS
4571  3A  27  4A          13.0                 LDA     SUNIT
4574  D3  E0               10.0                OUT     MBSEL

4576  3E  04                7.0                MVI     A,P.RPEN
4578  D3  4C               10.0                OUT     PENAB
457A  3E  60                7.0                MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
457C  D3  48               10.0                OUT     PDIAG           ;LCS, LWR
457E  AF                    4.0                XRA     A
457F  D3  44               10.0                OUT     TAMT
4581  D3  D2               10.0                OUT     TRKENA          ;CLEAR ALL TRACKS FROM
4583  D3  D2               10.0                OUT     TRKENA          ;CLEAR ALL TRACKS FROM
4585  32  24  4A          13.0                 STA     INDAT           ;CLEAR THE INPUT DATA BYTE
4588                                    ROUT    ROSH
 (1) 4588  D3  88           10.0                OUT     ROSH                    ;WRITE AC INTO ROSH
 (1) 458A  7F                4.0                MOV     A,A                     ;RETRY LINK
```

```
2208  458B  3E  A8        7.0  TST02C: MVI   A,R.PLO1.R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2209                                                                 ;SET TIE BUS JAM AND PLO DISABLE
2210  458D  D3  09       10.0          OUT   RPCTL
2211  458F  D3  08       10.0          OUT   RFIFOL                 ;CLOCK THE FIFO'S
2212  4591  3E  0D        7.0          MVI   A,RCLRT                ;ISSUE CLEAR ALL COMMAND
2213  4593  D3  08       10.0          OUT   RCMD
2214
2215  4595  00           4.0          NOP                          ;WAIT
2216  4596  00           4.0          NOP
2217  4597  00           4.0          NOP
2218  4598  00           4.0          NOP
2219
2220  4599  3E  A9        7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2221                                                                 ;SET TIE BUS JAM AND PLO DISABLE
2222  459B  D3  09       10.0          OUT   RPCTL
2223  459D  CD  50  48   18.0          CALL  CLECC                  ;CLEAR THE ECC CHARACTER
2224  45A0  16  07        7.0          MVI   D,@7                   ;SET UP THE LOOP COUNT
2225  45A2  3A  24  4A   13.0  TST02F: LDA   INDAT                  ;GET THE INPUT DATA
2226  45A5  CD  55  48   18.0          CALL  ECC                    ;CALCULATE ECC FOR
2227  45A8  15           4.0          DCR   D                      ;THE 7 INPUT BYTES
2228  45A9  C2  A2  45   10.0          JNZ   TST02F
2229
2230  45AC  0E  03        7.0          MVI   C,@3                   ;INIT THE LOOP COUNT
2231  45AE  3A  24  4A   13.0  TST02I: LDA   INDAT                  ;GET THE INPUT DATA
2232  45B1                       ROUT  R05L
 (1)  45B1  D3  8A       10.0                OUT   R05L                   ;WRITE AC INTO R05L
 (1)  45B3  7F           4.0                 MOV   A,A                    ;RETRY LINK
2233  45B4  32  40  48   13.0          STA   TRNIN                  ;FILL THE TRANSLATOR
2234  45B7  32  41  48   13.0          STA   TRNIN+1                ;SUBROUTINE INPUT
2235  45BA  32  42  48   13.0          STA   TRNIN+2                ;BUFFER
2236  45BD  32  43  48   13.0          STA   TRNIN+3                ;
2237  45C0  CD  C0  47   18.0          CALL  T4X5                   ;TRANSLATE THE SUBGROUP
2238
2239  45C3  21  44  48   10.0          LXI   H,TRNOUT               ;GET POINTER TO TRANSLATE DATA TABLE
2240  45C6  06  05        7.0          MVI   B,@5                   ;SET UP LOOP COUNT
2241  45C8  7E            7.0  TST02G: MOV   A,M                    ;GET A DATA BYTE
2242  45C9  D3  40       10.0          OUT   TCMD                   ;STORE DATA IN COMMAND ADDRESS
2243  45CB  23            6.0          INX   H                      ;POINT TO DATA PARITY
2244  45CC  7E            7.0          MOV   A,M                    ;GET THE DATA PARITY
2245
2246  45CD  07            4.0          RLC                          ;POSITION FOR OUTPUT
2247  45CE  F6  60        7.0          ORI   P.LWR!P.LCS            ;OR IN CONTROL BITS
2248
2249  45D0  D3  48       10.0          OUT   PDIAG                  ;OUTPUT THE DATA PARITY
2250
2251  45D2  D3  08       10.0          OUT   RFIFOL                 ;CLOCK DATA INTO THE FIFO'S
2252
2253  45D4  23            6.0          INX   H                      ;UPDATE THE TABLE POINTER
2254  45D5  05            4.0          DCR   B                      ;DECREMENT LOOP COUNT
2255  45D6  C2  C8  45   10.0          JNZ   TST02G                 ;DO UNTIL LOOP COUNT - 0
2256
2257  45D9  3A  24  4A   13.0          LDA   INDAT                  ;GET THE INPUT DATA
2258  45DC  32  40  48   13.0          STA   TRNIN                  ;FILL THE FIRST THREE
2259  45DF  32  41  48   13.0          STA   TRNIN+1                ;BYTES OF THE TRANSLATOR
```

```
2260  45E2  32  42  48   13.0          STA   TRNIN+2          ;SUBROUTINE WITH THE INPUT DATA
2261  45E5  3A  78  48   13.0          LDA   ECCCHR           ;STORE THE CALCULATED
2262  45E8  32  43  48   13.0          STA   TRNIN+3          ;ECC CHARACTER AS THE LAST CHARACTER
2263
2264  45EB  CD  C0  47   18.0          CALL  T4X5             ;TRANSLATE THE SECOND SUBGROUP
2265
2266  45EE  21  44  48   10.0          LXI   H,TRNOUT         ;GET THE POINTER TO INPUT DATA TABLE
2267  45F1  06  05        7.0          MVI   B,@5             ;SET UP THE LOOP COUNT
2268  45F3  7E            7.0  TST02H: MOV   A,M             ;GET A DATA BYTE
2269  45F4  D3  40       10.0          OUT   TCMD             ;STORE DATA IN CMD ADDRESS
2270  45F6  23            6.0          INX   H                ;POINT TO DATA PARITY BIT
2271  45F7  7E            7.0          MOV   A,M              ;GET THE DATA PARITY
2272  45F8  07            4.0          RLC                    ;POSITION FOR OUTPUT
2273  45F9  F6  60        7.0          ORI   P.LWR!P.LCS      ;OR IN CONTROL BITS
2274  45FB  D3  48       10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
2275  45FD  D3  08       10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFOS
2276  45FF  23            6.0          INX   H                ;UPDATE THE TABLE POINTER
2277  4600  05            4.0          DCR   B                ;DECREMENT THE LOOP COUNT
2278  4601  C2  F3  45   10.0          JNZ   TST02H           ;DO UNTIL LOOP COUNT=0
2279
2280  4604  0D            4.0          DCR   C                ;PUT THE GROUP IN
2281  4605  C2  AE  45   10.0          JNZ   TST02I           ;THE FIFO TWICE
2282
2283  4608  3E  0B        7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
2284  460A  D3  0B       10.0          OUT   RCMD             ;COMMAND
2285  460C  11  01  00   10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
2286  460F  21  A8  FD   10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
2287  4612  D3  0C       10.0  2$:     OUT   RINST            ;STEP THE READ PATH
2288  4614  DB  01       10.0          IN    RPCHI            ;DATA READY SET?
2289  4616  E6  10        7.0          ANI   R.DRDY
2290  4618  C2  2D  46   10.0          JNZ   TST02A           ;YES-GO PROCESS
2291  461B  19           10.0          DAD   D                ;WATCH DOG TIMEOUT?
2292  461C  D2  12  46   10.0          JNC   2$               ;NO-CONTINUE
2293  461F                             ERR   TST02C,DUMMU,1
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  461F  CD  09  28   18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)       000A                              MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4622  0A                                .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4623  01                                .BYTE 1
 (1)  4624  CD  15  28   18.0          DUMMU:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4627  DA  8B  45   10.0                  JC    TST02C           ;LOOP ADDRESS IF LOOP SPECIFIED
2294                          ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2295                          ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2296                          ;<FATAL ERROR - MICRO TEST ABORTED
2297  462A  C3  92  47   10.0          JMP   EXIT
2298
2299  462D  00            4.0  TST02A: NOP
2300  462E  21  74  49   10.0          LXI   H,TIETBL
2301  4631  3A  24  4A   13.0          LDA   INDAT            ;GET THE INPUT DATA
2302  4634  E6  0F        7.0          ANI   @17              ;GET LO 4 BITS
2303  4636  CA  3F  46   10.0          JZ    2$               ;EXIT IF ZERO
2304  4639  23            6.0  1$:     INX   H                ;UPDATE TABLE POINTER
2305  463A  23            6.0          INX   H                ;
2306  463B  3D            4.0          DCR   A                ;DECREMENT LOOP COUNT
```

```
2307  463C  C2  39  46    10.0          JNZ   1$              ;CONTINUE UNTIL ZERO
2308  463F  7E              7.0  2$:     MOV   A,M             ;GET HIGH BYTE FROM TABLE
2309  4640  57              4.0          MOV   D,A             ;MOVE TO D
2310  4641  23              6.0          INX   H               ;GET LOW BYTE FROM TABLE
2311  4642  7E              7.0          MOV   A,M             ;
2312  4643  5F              4.0          MOV   E,A             ;MOVE TO E
2313  4644  EB              4.0          XCHG                  ;MOV D+E TO H+L
2314  4645  06  08           7.0          MVI   B,$8            ;INIT THE LOOP COUNT
2315  4647  00              4.0  TST02B: NOP
2316  4648  7E              7.0          MOV   A,M             ;GET THE TIE BUS POINTER
2317  4649  D3  0A          10.0          OUT   RTIEB           ;LOAD THE TIE BUS
2318  464B  23              6.0          INX   H               ;UPDATE THE TABLE POINTER
2319  464C  DB  17          10.0          IN    RDATA           ;INPUT THE ACTUAL DATA BYTE
2320  464E  32  2A  4A      13.0          STA   DATAA           ;STORE IT
2321  4651  DB  15          10.0          IN    RPSTA           ;INPUT THE ACTUAL PARITY BIT
2322  4653  E6  40           7.0          ANI   R.DATA          ;REMOVE JUNK BITS
2323  4655  07              4.0          RLC                   ;POSITION THE BIT FOR COMPARE
2324  4656  07              4.0          RLC
2325  4657  32  2B  4A      13.0          STA   DATAAP          ;STORE IT
2326  465A  3A  24  4A      13.0          LDA   INDAT           ;GET THE EXPECTED DATA BYTE
2327  465D  32  2C  4A      13.0          STA   DATAE           ;STORE IT
2328  4660  A7              4.0          ANA   A               ;SET CONDITION BITS
2329  4661  E2  69  46      10.0          JPO   1$              ;ODD # OF ONES - CLEAR PARITY
2330  4664  3E  01           7.0          MVI   A,@1            ;EVEN # OF ONES - SET PARITY
2331  4666  C3  6A  46      10.0          JMP   2$
2332  4669  AF              4.0  1$:     XRA   A
2333  466A  32  2D  4A      13.0  2$:     STA   DATAEP
2334  466D  CD  8B  48      18.0          CALL  CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
2335  4670  DA  8B  45      10.0          JC    TST02C          ;IF LOOP ON ERROR REPEAT THE TEST
2336  4673  05              4.0          DCR   B               ;DECREMENT THE LOOP COUNT
2337  4674  D3  0C          10.0          OUT   RINST           ;SINGLE STEP THE READ PATH
2338  4676  C2  47  46      10.0          JNZ   TST02B          ;DO UNTIL LOOP COUNT = ZERO
2339  4679  7E              7.0          MOV   A,M             ;GET THE LAST POINTER
2340  467A  D3  0A          10.0          OUT   RTIEB           ;LOAD THE TIE BUS
2341  467C  DB  17          10.0          IN    RDATA           ;INPUT THE ACTUAL ECC CHARACTER
2342  467E                                ROUT  ADATA           ;STORE IN CAS
 (1)  467E  D3  94          10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  4680  7F              4.0                 MOV   A,A               ;RETRY LINK
2343  4681  47              4.0          MOV   B,A             ;SAVE IN REG. B
2344  4682  3A  78  48      13.0          LDA   ECCCHR          ;GET THE CORRECT ECC CHARACTER
2345  4685                                ROUT  EDATA           ;STORE IN CAS
 (1)  4685  D3  95          10.0                OUT   EDATA             ;WRITE AC INTO EDATA
 (1)  4687  7F              4.0                 MOV   A,A               ;RETRY LINK
2346  4688  B8              4.0          CMP   B               ;ACTUAL=EXPECTED?
2347  4689  CA  91  46      10.0          JZ    ECCCK2          ;YES-CONTINUE
```

```
2349  468C                                          ERRB    TST02C,ECCCK2,1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  468C  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  468F  08                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4690  01                                              .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4691  CD  15  28        18.0                  ECCCK2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4694  DA  8B  45        10.0                          JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2350                                        ;>ECC CHARACTER INCORRECT AT OUTPUT OF M8950'S
2351                                        ;<
2352  4697  11  01  00        10.0          LXI     D,1             ;SET WATCHDOG TIMER INCREMENT
2353  469A  21  A8  FD        10.0          LXI     H,-600          ;SET WATCHDOG COUNT TO 600
2354  469D  D3  0C             10.0  3$:     OUT     RINST           ;SINGLE STEP THE READ PATH
2355  469F  DB  01             10.0          IN      RPCHI           ;DATA READY SET?
2356  46A1  E6  10              7.0          ANI     R.DRDY
2357  46A3  C2  B8  46         10.0          JNZ     TST02K          ;YES-GO PROCESS
2358  46A6  19                  10.0          DAD     D               ;NO-WATCHDOG TIMEOUT?
2359  46A7  D2  9D  46         10.0          JNC     3$              ;NO-CONTINUE
2360  46AA                                    ERR     TST02C,DUMMV,1
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46AA  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46AD  0C                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46AE  01                                              .BYTE   1
 (1)  46AF  CD  15  28        18.0                  DUMMV:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46B2  DA  8B  45        10.0                          JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2361                                        ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2362                                        ;>SENT TO THE ECC
2363                                        ;<FATAL ERROR - MICRO TEST ABORTED
2364  46B5  C3  92  47         10.0          JMP     EXIT
2365
2366  46B8  16  01              7.0  TST02K: MVI     D,@1                    ;INIT THE LOOP COUNT
2367  46BA  AF                  4.0  TST02Y: XRA     A
2368  46BB                               ROUT    R05H
 (1)  46BB  D3  8B             10.0                  OUT     R05H                    ;WRITE AC INTO R05H
 (1)  46BD  7F                  4.0                  MOV     A,A                     ;RETRY LINK
2369  46BE  3A  24  4A         13.0          LDA     INDAT                   ;GET THE INPUT DATA
2370  46C1                               ROUT    EDATA
 (1)  46C1  D3  95             10.0                  OUT     EDATA                   ;WRITE AC INTO EDATA
 (1)  46C3  7F                  4.0                  MOV     A,A                     ;RETRY LINK
2371  46C4  A7                  4.0          ANA     A                       ;SET CONDITION BITS
2372  46C5  3E  00              7.0          MVI     A,0                     ;CLEAR THE ACC.
2373  46C7  E2  CB  46         10.0          JPO     1$                      ;ODD # OF ONES - CONTINUE
2374  46CA  3C                  4.0          INR     A                       ;EVEN # OF ONES - SET PARITY
2375  46CB  32  25  4A         13.0  1$:     STA     INDATP                  ;STORE THE PARITY BIT
2376  46CE  D3  0C             10.0  TST02L: OUT     RINST                   ;SINGLE STEP THE READ LOGIC
2377  46D0  3A  24  4A         13.0          LDA     INDAT
2378  46D3  47                  4.0          MOV     B,A
2379  46D4  DB  19             10.0          IN      ECCCOR
2380  46D6  B8                  4.0          CMP     B
2381  46D7  CA  E6  46         10.0          JZ      TST02M
2382  46DA                               ROUT    ADATA
 (1)  46DA  D3  94             10.0                  OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  46DC  7F                  4.0                  MOV     A,A                     ;RETRY LINK
```

```
2383  46DD  78                    4.0          MOV     A,B             ;GET THE BYTE COUNT
2384  46DE                                     ROUT    R05L            ;STORE IN CAS
 (1)  46DE  D3  8A                10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  46E0  7F                    4.0                  MOV     A,A             ;RETRY LINK
2385  46E1                                     ERRB    TST02C,TST02M,@1
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46E1  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000D                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E4  0D                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46E5  01                                         .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  46E6  CD  15  28            18.0         TST02M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46E9  DA  8B  45            10.0                 JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2386                                     ;>CORRECTED DATA FROM THE ECC-INCORRECT
2387                                     ;>NO CORRECTION SHOULD HAVE TAKEN PLACE
2388
2389  46EC  DB  15                10.0          IN      RPSTA           ;GET THE ECC PARITY BIT
2390  46EE  E6  80                 7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2391  46F0  07                     4.0          RLC                     ;POSITION BIT FOR COMPARE
2392  46F1  4F                     4.0          MOV     C,A             ;SAVE IN REG. C
2393  46F2  3A  25  4A             13.0         LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2394  46F5  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2395  46F6  CA  11  47             10.0         JZ      TST02QO         ;GO CHECK FOR PARITY-0
2396  46F9  79                     4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
2397  46FA  FE  01                 7.0          CPI     @1
2398  46FC  CA  08  47             10.0         JZ      TST02RG
2399  46FF  78                     4.0          MOV     A,B
2400  4700                                     ROUT    R05L
 (1)  4700  D3  8A                10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  4702  7F                    4.0                  MOV     A,A             ;RETRY LINK
2401  4703                                     ERR     TST02C,TST02RG,@1
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4703  CD  09  28            18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000E                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4706  0E                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4707  01                                         .BYTE   @1
 (1)  4708  CD  15  28            18.0         TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4708  DA  8B  45            10.0                 JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2402                                     ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2403                                     ;<
2404
2405  470E  C3  25  47             10.0         JMP     TST02SZ         ;CONTINUE WITH TEST
2406  4711  79                     4.0  TST02QO: MOV     A,C             ;CHECK FOR PARITY = ZERO
2407  4712  A7                     4.0          ANA     A               ;SET CONDITION BITS
2408  4713  CA  1F  47             10.0         JZ      TST02TP         ;CONTINUE IF ZERO
2409  4716  78                     4.0          MOV     A,B
2410  4717                                     ROUT    R05L
 (1)  4717  D3  8A                10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  4719  7F                    4.0                  MOV     A,A             ;RETRY LINK
```

```
2412  471A                                          ERR     TST02C,TST02TP,@1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  471A  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000F                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  471D  0F                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  471E  01                                              .BYTE   @1
 (1)  471F  CD  15  28        18.0                  TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4722  DA  88  45        10.0                          JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2413                                        ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2414                                        ;<
2415
2416  4725  7A                   4.0  TST02SZ: MOV     A,D             ;GET THE CHARACTER COUNT
2417  4726  3C                   4.0          INR     A               ;INCREMENT
2418  4727  FE  08               7.0          CPI     $8              ;DONE?
2419  4729  CA  32  47          10.0          JZ      TST02N          ;YES-GO CHECK ECC
2420  472C  57                   4.0          MOV     D,A             ;NO-CONTINUE
2421  472D  00                   4.0          NOP
2422  472E  00                   4.0          NOP
2423  472F  C3  BA  46          10.0          JMP     TST02Y
2424
2425  4732  D3  0C               10.0  TST02N: OUT     RINST           ;SINGLE STEP THE READ PATH
2426  4734  3A  78  48          13.0          LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
2427  4737                               ROUT    EDATA           ;SAVE IN CAS
 (1)  4737  D3  95               10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4739  7F                   4.0                   MOV     A,A             ;RETRY LINK
2428  473A  47                   4.0          MOV     B,A             ;SAVE IN REGISTER B
2429  473B  DB  19               10.0          IN      ECCCOR
2430  473D                               ROUT    ADATA
 (1)  473D  D3  94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  473F  7F                   4.0                   MOV     A,A             ;RETRY LINK
2431  4740  B8                   4.0          CMP     B
2432  4741  CA  4F  47          10.0          JZ      TST02U
2433  4744  3A  24  4A          13.0          LDA     INDAT           ;GET THE DATA/COUNT
2434  4747                               ROUT    R05L            ;SAVE IT
 (1)  4747  D3  8A               10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  4749  7F                   4.0                   MOV     A,A             ;RETRY LINK
2435  474A                               ERR     TST02C,TST02U,1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  474A  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0010                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  474D  10                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  474E  01                                              .BYTE   1
 (1)  474F  CD  15  28        18.0                  TST02U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4752  DA  8B  45        10.0                          JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2436                                        ;>CORRECTED ECC CHARACTER INCORRECT
2437                                        ;<
2438  4755  DB  1A               10.0          IN      ECCSTA          ;GET THE ECC STATUS
2439  4757  E6  0F                7.0          ANI     @17             ;REMOVE UNWANTED BITS
2440  4759  CA  6E  47          10.0          JZ      TST02V
2441  475C                               ROUT    ADATA
 (1)  475C  D3  94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  475E  7F                   4.0                   MOV     A,A             ;RETRY LINK
2442  475F  AF                   4.0          XRA     A
2443  4760                               ROUT    EDATA
```

```
  (1) 4760 D3 95          10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
  (1) 4762 7F              4.0                    MOV     A,A             ;RETRY LINK
2444 4763 3A 24 4A        13.0            LDA     INDAT           ;GET THE DATA/COUNT
2445 4766                                 ROUT    R05L            ;SAVE IT
  (1) 4766 D3 8A          10.0                    OUT     R05L            ;WRITE AC INTO R05L
  (1) 4768 7F              4.0                    MOV     A,A             ;RETRY LINK
2446 4769                                 ERRB    TST02C,TST02V,1
  (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
  (1) 4769 CD 12 28       18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)      0011                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1) 476C 11                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1) 476D 01                                             .BYTE   1               ;PRINT ROUTINE NUMBER
  (1) 476E CD 15 23       18.0                    TST02V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1) 4771 DA 88 45       10.0                            JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2447                                    ;>ECC STATUS INCORRECT AFTER A DATA GROUP
2448                                    ;<
2449
2450 4774 3A 24 4A        13.0            LDA     INDAT
2451 4777 3C              4.0            INR     A
2452 4778 32 24 4A        13.0            STA     INDAT
2453 477B C2 88 45        10.0            JNZ     TST02C
2454 477E                                 ENDTST  TST02X
  (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2) 477E                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2) 477E CD 06 28       18.0                            CALL    REQST
  (2) 4781 00                                             .BYTE                   ;DATA PATTERN NUMBER
  (2) 4782 00 00                                          .WORD                   ;SYSTEM '' COUNT
  (2) 4784 00 00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2) 4786 00                                             .BYTE                   ;DATA COMPARE FLAG IF =1
  (2) 4787 07                                             .BYTE   7               ;REQUEST CODE
  (1) 4788 3A 9A 4F       13.0            LDA     ITERA           ;GET ITERATION COUNT
  (1) 478B 3D              4.0            DCR     A               ;DOWNCOUNT
  (1) 478C 32 9A 4F       13.0            STA     ITERA           ;SAVE COUNT
  (1) 478F F2 66 45       10.0            JP      TST02X          ;DO TEST UNTIL TILL = 0
2455 4792 C3 18 28        10.0    EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*****************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*----------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*****************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*****************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*****************************************************************

                                12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                                12.0          PUSH    B
                                 7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
                                10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
                                 7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                                 4.0          ORA     B               ;ADD IN THE SELECTED PORT #
                                10.0          OUT     MBSEL           ;RESET TO THIS PORT
                                 7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                                10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                                 4.0          XRA     A               ;CLEAR TU COMMAND A
                                10.0          OUT     TCMD            ;
                                 7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                                10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                                 7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
                                10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
                                 4.0          XRA     A
                                10.0          OUT     TAMT            ;CLEAR AMTIE WORD
                                10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                                10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
                                 4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                                 4.0          MOV     A,B
                                 7.0          CPI     4               ;DONE?
                                10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                                10.0          POP     B               ;RESET B & C
                                10.0          POP     PSW             ;ALL DONE
                                10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
47C0                                    SSUB
(1)                                     ; ************************************************************************
(1)                                     ;*SUBROUTINE TITLE
(1)                                     ;*---------- -----
                                        ;*4X5 TRANSLATE A SUBGROUP
47C0                                    SD
(1)                                     ; ************************************************************************
(1)                                     ;*DESCRIPTION
(1)                                     ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                        ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
47C0                                    SP
(1)                                     ; ************************************************************************
(1)                                     ;*PROCEDURE
(1)                                     ;*---------
                                        ;*BGNSUB
                                        ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                        ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER 'OUTPUT BUFFER IS 9X5)
                                        ;*    SET LOOP COUNT TO 9
                                        ;*    BGNDO
                                        ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                        ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                        ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                        ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                        ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                        ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                        ;*    :        TO BUILD THE INDEX)
                                        ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                        ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                        ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                        ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                        ;*    :        DATA BYTE.
                                        ;*    :    DECREMENT THE LOOP COUNT
                                        ;*    :    DO UNTIL THE LOOP COUNT = 0
                                        ;*    ENDDO
                                        ;*ENDSUB
                                        ;*
47C0                                    S
(1)                                     ; ************************************************************************

47C0  F5              12.0  T4X5:   PUSH    PSW             ;SAVE REGISTER A + PSW
47C1  C5              12.0          PUSH    B               ;SAVE REGISTER B + C
47C2  D5              12.0          PUSH    D               ;SAVE REGISTER D + E
47C3  E5              12.0          PUSH    H               ;SAVE REGISTER H + L
                                    ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                                    ;CLEAR THE TRANSLATED DATA TABLE.
47C4  06  0A           7.0          MVI     B,10            ;SET UP LOOP COUNT
47C6  11  44  48      10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
47C9  AF               4.0          XRA     A               ;CLEAR A
47CA  12               7.0  D4X5:   STAX    D               ;STORE IN THE TABLE
47CB  13               6.0          INX     D               ;UPDATE TABLE POINTER
47CC  05               4.0          DCR     B               ;DECREMENT LOOP COUNT
47CD  C2  CA  47      10.0          JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
```

```
2546  47D0  0E 09        7.0          MVI   C,9              ;SET UP TRACK COUNT
2547  47D2  06 04        7.0  B4X5:   MVI   B,4              ;SET UP BIT COUNT
2548  47D4  11 40 48     10.0         LXI   D,TRNIN
2549  47D7  AF           4.0          XRA   A                ;CLEAR THE GROUP POSITION COUNT
2550  47D8  32 4E 48     13.0         STA   GP4X5
2551  47DB  1A           7.0  C4X5:   LDAX  D                ;GET A DATA BYTE
2552  47DC  A7           4.0          ANA   A                ;SET CONDITION BITS
2553  47DD  E2 E1 47     10.0         JPO   PO4X5            ;ODD PARITY LEAVE CARRY CLEAR
2554  47E0  37           4.0          STC                    ;EVEN PARITY SET CARRY
2555  47E1  1F           4.0  PO4X5:  RAR                    ;SHIFT OUT DESIRED BIT
2556  47E2  12           7.0          STAX  D                ;STORE RESULT BACK IN TEMP TABLE
2557  47E3  3A 4E 48     13.0         LDA   GP4X5            ;GET THE GROUP POSITION BYTE
2558  47E6  17           4.0          RAL                    ;PUT IN THIS DATA BIT
2559  47E7  32 4E 48     13.0         STA   GP4X5            ;SAVE THE UPDATED GROUP POSITION BYTE
2560  47EA  13           6.0          INX   D                ;UPDATE THE TABLE POINTER
2561  47EB  05           4.0          DCR   B                ;DECREMENT THE BIT COUNT
2562  47EC  C2 DB 47     10.0         JNZ   C4X5             ;DO UNTIL ALL 4 BITS ARE RECEIVED
2563
2564                                ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2565
2566                                ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2567  47EF  06 05        7.0          MVI   B,5              ;SET UP LOOP COUNT
2568  47F1  11 44 48     10.0         LXI   D,TRNOUT         ;GET POINTER TO TRANSLATED DATA TABLE
2569  47F4  13           6.0  E4X5:   INX   D                ;POINT TO PARITY BIT
2570  47F5  1A           7.0          LDAX  D                ;GET PARITY BIT
2571  47F6  A7           4.0          ANA   A                ;CLEAR THE CARRY BIT
2572  47F7  1F           4.0          RAR                    ;SHIFT TO THE CARRY BIT
2573  47F8  12           7.0          STAX  D                ;STORE IT BACK
2574  47F9  1B           6.0          DCX   D                ;DECREMENT TO DATA BITS
2575  47FA  1A           7.0          LDAX  D                ;GET THE DATA BITS
2576  47FB  1F           4.0          RAR                    ;SHIFT IN CARRY BITS
2577  47FC  12           7.0          STAX  D                ;STORE BACK IN TABLE
2578  47FD  13           6.0          INX   D                ;POINT TO NEXT TABLE ENTRY
2579  47FE  13           6.0          INX   D                ;
2580  47FF  05           4.0          DCR   B                ;DECREMENT LOOP COUNT
2581  4800  C2 F4 47     10.0         JNZ   E4X5             ;DO UNTIL LOOP COUNT=0
2582  4803  11 30 48     10.0         LXI   D,TAB4X5         ;LOAD ADDRESS OF TRANSLATION TABLE
2583  4806  26 00        7.0          MVI   H,0              ;CLEAR REGISTER H
2584  4808  3A 4E 48     13.0         LDA   GP4X5            ;GET GROUP POSITION COUNT
2585  480B  6F           4.0          MOV   L,A
2586  480C  19           10.0         DAD   D                ;ADD GROUP POSITION COUNT TO TABLE
2587  480D  7E           7.0          MOV   A,M              ;GET TRANSLATED DATA
2588  480E  32 4F 48     13.0         STA   TRNTMP
2589                                ;''TRNTMP'' = THE TRANSLATED DATA
2590
2591  4811  11 44 48     10.0         LXI   D,TRNOUT         ;GET POINTER TO TRANSLATED DATA TABLE
2592  4814  06 05        7.0          MVI   B,5              ;SET UP LOOP COUNT
2593  4816  3A 4F 48     13.0  F4X5:   LDA   TRNTMP           ;GET TRANSLATED DATA
2594  4819  13           6.0          INX   D                ;POINT TO PARITY POSITION
2595  481A  A7           4.0          ANA   A                ;CLEAR CARRY BIT
2596  481B  1F           4.0          RAR                    ;SHIFT OUT A BIT
2597  481C  32 4F 48     13.0         STA   TRNTMP           ;STORE TRANSLATED DATA
2598  481F  1A           7.0          LDAX  D                ;GET THE PARITY BIT (ALWAYS ZERO)
2599  4820  17           4.0          RAL                    ;ROLL IN THE CARRY BIT
```

```
2600  4821  12              7.0          STAX   D              ;STORE AS THE PARITY BIT
2601  4822  13              6.0          INX    D              ;UPDATE TABLE POINTER
2602  4823  05              4.0          DCR    B              ;DECREMENT THE LOOP COUNT
2603  4824  C2  16  48      10.0          JNZ    F4X5           ;DO UNTIL LOOP COUNT=0
2604
2605                                  ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2606
2607  4827  0D               4.0          DCR    C              ;DECREMENT THE TRACK COUNT
2608  4828  C2  D2  47      10.0          JNZ    B4X5           ;DO UNTIL ALL TRACKS TRANSLATED
2609  482B  E1              10.0          POP    H              ;RESTORE REGISTER H + L
2610  482C  D1              10.0          POP    D              ;RESTORE REGISTER D + E
2611  482D  C1              10.0          POP    B              ;RESTORE REGISTER B + C
2612  482E  F1              10.0          POP    PSW            ;RESTORE REGISTER A + PSW
2613  482F  C9              10.0          RET                   ;RETURN TO USER
2614                                  .SBTTL  TABLE 4 X 5 TRANSLATION
2615
2616                                  ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2617                                  ;   INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2618                                  ;    1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
2619                                  ;
2620                                  ;         0000                              10011
2621                                  ;         0001                              11011
2622                                  ;         0010                              01001
2623                                  ;         0011                              11001
2624                                  ;         0100                              10111
2625                                  ;         0101                              10101
2626                                  ;         0110                              01101
2627                                  ;         0111                              11101
2628                                  ;         1000                              01011
2629                                  ;         1001                              10010
2630                                  ;         1010                              01010
2631                                  ;         1011                              11010
2632                                  ;         1100                              01111
2633                                  ;         1101                              10110
2634                                  ;         1110                              01110
2635                                  ;         1111                              11110
2636
2637  4830  13                TAB4X5: .BYTE  @23
2638  4831  1B                        .BYTE  @33
2639  4832  09                        .BYTE  @11
2640  4833  19                        .BYTE  @31
2641  4834  17                        .BYTE  @27
2642  4835  15                        .BYTE  @25
2643  4836  0D                        .BYTE  @15
2644  4837  1D                        .BYTE  @35
2645  4838  0B                        .BYTE  @13
2646  4839  12                        .BYTE  @22
2647  483A  0A                        .BYTE  @12
2648  483B  1A                        .BYTE  @32
2649  483C  0F                        .BYTE  @17
2650  483D  16                        .BYTE  @26
2651  483E  0E                        .BYTE  @16
2652  483F  1E                        .BYTE  @36
2653
```

```
2655                                    .SBTTL  SUBROUTINE VARIABLES
2656
2657  4840 0004                         TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2658                                                                    ;BE TRANSLATED
2659  4844 000A                         TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2660                                                                    ;TRANSLATION
2661  484E   00                         GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2662                                                                    ;BE TRANSLATED
2663  484F   00                         TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2664
2665                                    .SBTTL  SUBROUTINE CLEAR ECC
2666  4850                              S
 (1)                                    ; ***************************************************************************
2667                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
2668                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2669                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2670  4850                              S
 (1)                                    ; ***************************************************************************
2671
2672  4850  AF                   4.0    CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
2673  4851  32  78  48           13.0            STA     ECCCHR          ;CLEAR THE ECC CHARACTER
2674  4854  C9                   10.0            RET                     ;RETURN TO USER
2675
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        ;
                                        ;*************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF 'ECCCHR'' TO UPDATE THE CONTENTS OF 'ECCCHR'' ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        ;
                                        ;*************************************************************************
                                        ECC:    LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
                                                XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
                                                MOV     E,A             ;SAVE XOR RESULT IN E
                                                ANI     $10             ;IS BIT #4 OF RESULT SET
                                                MOV     A,E             ;RESTORE XOR RESULT FROM B
                                                JZ      ECC1            ;CONTINUE IF BIT #4 RESET
                                                XRI     $23             ;ELSE-XOR WITH 23
                                        ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                                                        ;
                                                XRA     A               ;CLEAR A
                                                MOV     C,A             ;CLEAR THE TRANSLATE RESULT
                                                LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
                                                CALL    TRANS           ;TRANSLATE THE BITS
                                                MOV     A,C             ;GET THE TRANSLATED RESULT
                                                STA     ECCCHR          ;STORE RESULT
                                                RET
                                        ECCTBL: $OR                     ;BIT 0 = POSITION 3
                                                $20                     ;BIT 1 = POSITION 5
                                                $02                     ;BIT 2 = POSITION 1
                                                $40                     ;BIT 3 = POSITION 6
                                                $80                     ;BIT 4 = POSITION 7
                                                $01                     ;BIT 5 = POSITION 0
                                                $10                     ;BIT 6 = POSITION 4
                                                $04                     ;BIT 7 = POSITION 2
                                        E   CHR: .BY E  0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

4879  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
487B  7B                4.0  TRANS1: MOV   A,E        ;GET CHAR TO BE TRANSLATED
487C  A0                4.0          ANA   B          ;SEE IF BIT POSITION IN 'B' IS SET
487D  CA  83  48       10.0          JZ    TRANS2     ;DO NEXT BIT POSITION IF NOT SET
4880  79                4.0          MOV   A,C        ;GET PREVIOUS RESULT OF 'OR'
4881  B6                7.0          ORA   M          ;'OR' IN NEW POSITION
4882  4F                4.0          MOV   C,A        ;SAVE RESULT

4883  78                4.0  TRANS2: MOV   A,B
4884  07                4.0          RLC              ;POSITION MASK TO NEXT BIT
4885  47                4.0          MOV   B,A
4886  D8               12.0          RC               ;EXIT WHEN ALL POSITIONS DONE
4887  23                6.0          INX   H          ;POINT TO NEXT TABLE ENTRY
4888  C3  7B  48       10.0          JMP   TRANS1     ;PROCESS NEXT BIT
```

```
2735
2736  488B                      .SBTTL  SUBROUTINE CHECK 5X4
                                SSUB
                                ;*************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK 5X4 SUBROUTINE
2738  488B                      SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
2739                            ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ CHANNEL DATA
2740                            ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
2741                            ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
2742  488B                      SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
2743                            ;*BGNSUB
2744                            ;*     IF CHANNEL 0 ACTUAL INFORMATION NOT - CHANNEL 0 EXPECTED
2745                            ;*       THEN-ERROR CHANNEL 0
2746                            ;*       ELSE-CONTINUE
2747                            ;*     ENDIF
2748                            ;*     IF CHANNEL 1 ACTUAL INFORMATION NOT - CHANNEL 1 EXPECTED
2749                            ;*       THEN-ERROR CHANNEL 1
2750                            ;*       ELSE-CONTINUE
2751                            ;*     ENDIF
2752                            ;*     IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
2753                            ;*       THEN-ERROR CHANNEL 2
2754                            ;*       ELSE-CONTINUE
2755                            ;*     ENDIF
2756                            ;*     IF CHANNEL 3 ACTUAL INFORMATION NOT - CHANNEL 3 EXPECTED
2757                            ;*       THEN-ERROR CHANNEL 3
2758                            ;*       ELSE-CONTINUE
2759                            ;*     ENDIF
2760                            ;*     IF CHANNEL 4 ACTUAL INFORMATION NOT - CHANNEL 4 EXPECTED
2761                            ;*       THEN-ERROR CHANNEL 4
2762                            ;*       ELSE-CONTINUE
2763                            ;*     ENDIF
2764                            ;*     IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
2765                            ;*       THEN-ERROR CHANNEL 5
2766                            ;*       ELSE-CONTINUE
2767                            ;*     ENDIF
2768                            ;*     IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
2769                            ;*       THEN-ERROR CHANNEL 6
2770                            ;*       ELSE-CONTINUE
2771                            ;*     ENDIF
2772                            ;*     IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
2773                            ;*       THEN-ERROR CHANNEL 7
2774                            ;*       ELSE-CONTINUE
2775                            ;*     ENDIF
2776                            ;*     IF CHANNEL P ACTUAL INFORMATION NOT = CHANNEL P EXPECTED
2777                            ;*       THEN-ERROR CHANNEL P
2778                            ;*       ELSE-CONTINUE
2779                            ;*     ENDIF
```

```
2780
2781  488B                                          ;*ENDSUB
 (1)                                                $
                                                    ; ***********************************************************************************
2782  488B  E5                          12.0  CK5X4:  PUSH    H                 ;SAVE H&L
2783  488C  C5                          12.0          PUSH    B                 ;SAVE B&C
2784  488D  21   2C   4A                10.0          LXI     H,DATAE           ;LOAD EXPECTED DATA ADDRESS
2785  4890  3A   2A   4A                13.0          LDA     DATAA             ;GET THE ACTUAL DATA
2786  4893                                            ROUT    ADATA             ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  4893  D3   94                     10.0                  OUT     ADATA             ;WRITE AC INTO ADATA
 (1)  4895  7F                          4.0                   MOV     A,A               ;RETRY LINK
2787  4896  E6   01                     7.0           ANI     $01               ;REMOVE UNWANTED BITS
2788  4898  47                          4.0           MOV     B,A               ;SAVE IN B
2789  4899  7E                          7.0           MOV     A,M               ;GET EXPECTED DATA
2790  489A                                            ROUT    EDATA             ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  489A  D3   95                     10.0                  OUT     EDATA             ;WRITE AC INTO EDATA
 (1)  489C  7F                          4.0                   MOV     A,A               ;RETRY LINK
2791  489D  E6   01                     7.0           ANI     $01               ;REMOVE UNWANTED BITS
2792  489F  B8                          4.0           CMP     B                 ;COMPARE
2793  48A0  CA   A8   48                10.0          JZ      CKAM0             ;CONTINUE IF EQUAL
2794  48A3                                            ERRB    OUT,CKAM0,1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  48A3  CD   12   28                18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)       0012                                             MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48A6  12                                               .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  48A7  01                                               .BYTE   1                 ;PRINT ROUTINE NUMBER
 (1)  48A8  CD   15   28                18.0          CKAM0:: CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48AB  DA   71   49                10.0                  JC      OUT               ;LOOP ADDRESS IF LOOP SPECIFIED
2795                                            ;>CHANNEL 0 FAILED
2796  48AE  3A   2A   4A                13.0          LDA     DATAA             ;GET THE ACTUAL DATA
2797  48B1  E6   02                     7.0           ANI     $02               ;REMOVE UNWANTED BITS
2798  48B3  47                          4.0           MOV     B,A               ;SAVE IN B
2799  48B4  7E                          7.0           MOV     A,M               ;GET EXPECTED DATA
2800  48B5  E6   02                     7.0           ANI     $02               ;REMOVE UNWANTED BITS
2801  48B7  B8                          4.0           CMP     B                 ;COMPARE
2802  48B8  CA   C0   48                10.0          JZ      CKAM1             ;CONTINUE IF EQUAL
2803  48BB                                            ERRB    OUT,CKAM1,1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  48BB  CD   12   28                18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)       0013                                             MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48BE  13                                               .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  48BF  01                                               .BYTE   1                 ;PRINT ROUTINE NUMBER
 (1)  48C0  CD   15   28                18.0          CKAM1:: CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48C3  DA   71   49                10.0                  JC      OUT               ;LOOP ADDRESS IF LOOP SPECIFIED
2804                                            ;>CHANNEL 1 FAILED
2805  48C6  3A   2A   4A                13.0          LDA     DATAA             ;GET THE ACTUAL DATA
2806  48C9  E6   04                     7.0           ANI     $04               ;REMOVE UNWANTED BITS
2807  48CB  47                          4.0           MOV     B,A               ;SAVE IN B
2808  48CC  7E                          7.0           MOV     A,M               ;GET EXPECTED DATA
2809  48CD  E6   04                     7.0           ANI     $04               ;REMOVE UNWANTED BITS
2810  48CF  B8                          4.0           CMP     B                 ;COMPARE
2811  48D0  CA   D8   48                10.0          JZ      CKAM2             ;CONTINUE IF EQUAL
```

```
2813  48D3                                                  ERRB    OUT,CKAM2,1
 (1)                                                        ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  48D3   CD   12   28        18.0                              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0014                                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48D6   14                                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48D7   01                                                    .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  48D8   CD   15   28        18.0                      CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48DB   DA   71   49        10.0                              JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2814                                                    ;>CHANNEL 2 FAILED
2815  48DE   3A   2A   4A        13.0            LDA     DATAA           ;GET THE ACTUAL DATA
2816  48E1   E6   08              7.0            ANI     $08             ;REMOVE UNWANTED BITS
2817  48E3   47                   4.0            MOV     B,A             ;SAVE IN B
2818  48E4   7E                   7.0            MOV     A,M             ;GET EXPECTED DATA
2819  48E5   E6   08              7.0            ANI     $08             ;REMOVE UNWANTED BITS
2820  48E7   B8                   4.0            CMP     B               ;COMPARE
2821  48E8   CA   F0   48        10.0            JZ      CKAM3           ;CONTINUE IF EQUAL
2822  48EB                                        ERRB    OUT,CKAM3,1
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  48EB   CD   12   28        18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0015                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48EE   15                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48EF   01                                            .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  48F0   CD   15   28        18.0              CKAM3:: CALL    CKLOP           ;CHFCK LOOP FUNCTION - DO 2.2
 (1)  48F3   DA   71   49        10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2823                                                    ;>CHANNEL 3 FAILED
2824  48F6   3A   2A   4A        13.0            LDA     DATAA           ;GET THE ACTUAL DATA
2825  48F9   E6   10              7.0            ANI     $10             ;REMOVE UNWANTED BITS
2826  48FB   47                   4.0            MOV     B,A             ;SAVE IN B
2827  48FC   7E                   7.0            MOV     A,M             ;GET EXPECTED DATA
2828  48FD   E6   10              7.0            ANI     $10             ;REMOVE UNWANTED BITS
2829  48FF   B8                   4.0            CMP     B               ;COMPARE
2830  4900   CA   08   49        10.0            JZ      CKAM4           ;CONTINUE IF EQUAL
2831  4903                                        ERRB    OUT,CKAM4,1
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4903   CD   12   28        18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0016                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4906   16                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4907   01                                            .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4908   CD   15   28        18.0              CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4908   DA   71   49        10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2832                                                    ;>CHANNEL 4 FAILED
2833
2834  490E   3A   2A   4A        13.0            LDA     DATAA           ;GET THE ACTUAL DATA
2835  4911   E6   20              7.0            ANI     $20             ;REMOVE UNWANTED BITS
2836  4913   47                   4.0            MOV     B,A             ;SAVE IN B
2837  4914   7E                   7.0            MOV     A,M             ;GET EXPECTED DATA
2838  4915   E6   20              7.0            ANI     $20             ;REMOVE UNWANTED BITS
2839  4917   B8                   4.0            CMP     B               ;COMPARE
2840  4918   CA   20   49        10.0            JZ      CKAM5           ;CONTINUE IF EQUAL
```

```
2842  491B                                          ERRB    OUT,CKAM5,1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  491B  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0017                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  491E  17                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  491F  01                                              .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4920  CD  15  28        18.0                  CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4923  DA  71  49        10.0                          JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2843                                        ;>CHANNEL 5 FAILED
2844
2845  4926  3A  2A  4A        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2846  4929  E6  40               7.0          ANI     $40             ;REMOVE UNWANTED BITS
2847  492B  47                    4.0          MOV     B,A             ;SAVE IN B
2848  492C  7E                    7.0          MOV     A,M             ;GET EXPECTED DATA
2849  492D  E6  40               7.0          ANI     $40             ;REMOVE UNWANTED BITS
2850  492F  B8                    4.0          CMP     B               ;COMPARE
2851  4930  CA  38  49        10.0             JZ      CKAM6           ;CONTINUE IF EQUAL
2852  4933                                     ERRB    OUT,CKAM6,1
 (1)                                             ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4933  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0018                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4936  18                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4937  01                                              .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4938  CD  15  28        18.0                  CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  493B  DA  71  49        10.0                          JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2853                                        ;>CHANNEL 6 FAILED
2854
2855  493E  3A  2A  4A        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2856  4941  E6  80               7.0          ANI     $80             ;REMOVE UNWANTED BITS
2857  4943  47                    4.0          MOV     B,A             ;SAVE IN B
2858  4944  7E                    7.0          MOV     A,M             ;GET EXPECTED DATA
2859  4945  E6  80               7.0          ANI     $80             ;REMOVE UNWANTED BITS
2860  4947  B8                    4.0          CMP     B               ;COMPARE
2861  4948  CA  50  49        10.0             JZ      CKAM7           ;CONTINUE IF EQUAL
2862  494B                                     ERRB    OUT,CKAM7,1
 (1)                                             ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  494B  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0019                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  494E  19                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  494F  01                                              .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4950  CD  15  28        18.0                  CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4953  DA  71  49        10.0                          JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2863                                        ;>CHANNEL 7 FAILED
2864
2865  4956  21  2D  4A        10.0          LXI     H,DATAEP        ;GET THE EXPECTED DATA PARITY ADDRESS
2866  4959  3A  2B  4A        13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
2867  495C  E6  01               7.0          ANI     $01             ;REMOVE UNWANTED BITS
2868  495E  47                    4.0          MOV     B,A             ;SAVE IN B
2869  495F  7E                    7.0          MOV     A,M             ;GET EXPECTED DATA
2870  4960  E6  01               7.0          ANI     $01             ;REMOVE UNWANTED BITS
2871  4962  B8                    4.0          CMP     B               ;COMPARE
2872  4963  CA  68  49        10.0             JZ      CKAMP           ;CONTINUE IF EQUAL
```

```
2874  4966                                          ERRB    OUT,CKAMP,1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4966   CD   12   28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)         001A                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4969   1A                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  496A   01                                             .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  496B   CD   15   28        18.0                CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  496E   DA   71   49        10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2875                                    ;>CHANNEL P FAILED
2876
2877  4971   C1                 10.0  OUT:    POP     B               ;RESTORE B&C
2878  4972   E1                 10.0          POP     H               ;RESTORE H&L
2879  4973   C9                 10.0          RET
2880
```

```
2882                                    .SBTTL  TABLE TIE BUS VECTOR TABLE
2883
2884  4974  49  94                      TIETBL: .WORD   TI0000
2885  4976  49  9D                              .WORD   TI0001
2886  4978  49  A6                              .WORD   TI0010
2887  497A  49  AF                              .WORD   TI0011
2888  497C  49  B8                              .WORD   TI0100
2889  497E  49  C1                              .WORD   TI0101
2890  4980  49  CA                              .WORD   TI0110
2891  4982  49  D3                              .WORD   TI0111
2892  4984  49  DC                              .WORD   TI1000
2893  4986  49  E5                              .WORD   TI1001
2894  4988  49  EE                              .WORD   TI1010
2895  498A  49  F7                              .WORD   TI1011
2896  498C  4A  00                              .WORD   TI1100
2897  498E  4A  09                              .WORD   TI1101
2898  4990  4A  12                              .WORD   TI1110
2899  4992  4A  1B                              .WORD   TI1111
2900
```

```
                                .SBTTL  TABLE TIE BUS DATA VALUES

4994  01        TI0000: .BYTE   $01             ;TIE BUS FOR CLOCK 0
4995  04                .BYTE   $04             ;TIE BUS FOR CLOCK 1
4996  00                .BYTE   $00             ;TIE BUS FOR CLOCK 2
4997  00                .BYTE   $00             ;TIE BUS FOR CLOCK 3
4998  00                .BYTE   $00             ;TIE BUS FOR CLOCK 4
4999  00                .BYTE   $00             ;TIE BUS FOR CLOCK 5
499A  00                .BYTE   $00             ;TIE BUS FOR CLOCK 6
499B  00                .BYTE   $00             ;TIE BUS FOR CLOCK 7
499C  00                .BYTE   $00             ;TIE BUS FOR CLOCK 8

499D  00        TI0001: .BYTE   $00             ;TIE BUS FOR CLOCK 0
499E  00                .BYTE   $00             ;TIE BUS FOR CLOCK 1
499F  01                .BYTE   $01             ;TIE BUS FOR CLOCK 2
49A0  04                .BYTE   $04             ;TIE BUS FOR CLOCK 3
49A1  00                .BYTE   $00             ;TIE BUS FOR CLOCK 4
49A2  00                .BYTE   $00             ;TIE BUS FOR CLOCK 5
49A3  00                .BYTE   $00             ;TIE BUS FOR CLOCK 6
49A4  00                .BYTE   $00             ;TIE BUS FOR CLOCK 7
49A5  00                .BYTE   $00             ;TIE BUS FOR CLOCK 8

49A6  00        TI0010: .BYTE   $00             ;TIE BUS FOR CLOCK 0
49A7  00                .BYTE   $00             ;TIE BUS FOR CLOCK 1
49A8  00                .BYTE   $00             ;TIE BUS FOR CLOCK 2
49A9  00                .BYTE   $00             ;TIE BUS FOR CLOCK 3
49AA  01                .BYTE   $01             ;TIE BUS FOR CLOCK 4
49AB  04                .BYTE   $04             ;TIE BUS FOR CLOCK 5
49AC  00                .BYTE   $00             ;TIE BUS FOR CLOCK 6
49AD  00                .BYTE   $00             ;TIF BUS FOR CLOCK 7
49AE  00                .BYTE   $00             ;TIE BUS FOR CLOCK 8

49AF  00        TI0011: .BYTE   $00             ;TIE BUS FOR CLOCK 0
49B0  00                .BYTE   $00             ;TIE BUS FOR CLOCK 1
49B1  00                .BYTE   $00             ;TIE BUS FOR CLOCK 2
49B2  00                .BYTE   $00             ;TIE BUS FOR CLOCK 3
49B3  00                .BYTE   $00             ;TIE BUS FOR CLOCK 4
49B4  00                .BYTE   $00             ;TIE BUS FOR CLOCK 5
49B5  01                .BYTE   $01             ;TIE BUS FOR CLOCK 6
49B6  04                .BYTE   $04             ;TIE BUS FOR CLOCK 7
49B7  00                .BYTE   $00             ;TIE BUS FOR CLOCK 8

49B8  00        TI0100: .BYTE   $00             ;TIE BUS FOR CLOCK 0
49B9  00                .BYTE   $00             ;TIE BUS FOR CLOCK 1
49BA  00                .BYTE   $00             ;TIE BUS FOR CLOCK 2
49BB  00                .BYTE   $00             ;TIE BUS FOR CLOCK 3
49BC  00                .BYTE   $00             ;TIE BUS FOR CLOCK 4
49BD  00                .BYTE   $00             ;TIE BUS FOR CLOCK 5
49BE  00                .BYTE   $00             ;TIE BUS FOR CLOCK 6
49BF  01                .BYTE   $01             ;TIE BUS FOR CLOCK 7
49C0  04                .BYTE   $04             ;TIE BUS FOR CLOCK 8

49C1  02        TI0101: .BYTE   $02             ;TIE BUS FOR CLOCK 0
49C2  01                .BYTE   $01             ;TIE BUS FOR CLOCK 1
```

```
2956  49C3  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 2
2957  49C4  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 3
2958  49C5  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 4
2959  49C6  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 5
2960  49C7  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 6
2961  49C8  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 7
2962  49C9  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 8
2963
2964  49CA  00                TI0110: .BYTE   $00            ;TIE BUS FOR CLOCK 0
2965  49CB  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 1
2966  49CC  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 2
2967  49CD  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 3
2968  49CE  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 4
2969  49CF  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 5
2970  49D0  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 6
2971  49D1  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 7
2972  49D2  C0                        .BYTE   $00            ;TIE BUS FOR CLOCK 8
2973
2974  49D3  00                TI0111: .BYTE   $00            ;TIE BUS FOR CLOCK 0
2975  49D4  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 1
2976  49D5  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 2
2977  49D6  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 3
2978  49D7  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 4
2979  49D8  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 5
2980  49D9  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 6
2981  49DA  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 7
2982  49DB  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 8
2983
2984  49DC  00                TI1000: .BYTE   $00            ;TIE BUS FOR CLOCK 0
2985  49DD  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 1
2986  49DE  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 2
2987  49DF  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 3
2988  49E0  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 4
2989  49E1  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 5
2990  49E2  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 6
2991  49E3  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 7
2992  49E4  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 8
2993
2994  49E5  00                TI10C1: .BYTE   $00            ;TIE BUS FOR CLOCK 0
2995  49E6  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 1
2996  49E7  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 2
2997  49E8  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 3
2998  49E9  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 4
2999  49EA  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 5
3000  49EB  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 6
3001  49EC  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 7
3002  49ED  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 8
3003
3004  49EE  01                TI1010: .BYTE   $01            ;TIE BUS FOR CLOCK 0
3005  49EF  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 1
3006  49F0  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 2
3007  49F1  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 3
3008  49F2  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 4
3009  49F3  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 5
```

```
3010  49F4  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 6
3011  49F5  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 7
3012  49F6  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 8
3013
3014  49F7  00                TI1011: .BYTE   $00            ;TIE BUS FOR CLOCK 0
3015  49F8  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 1
3016  49F9  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 2
3017  49FA  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 3
3018  49FB  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 4
3019  49FC  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 5
3020  49FD  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 6
3021  49FE  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 7
3022  49FF  00                        .BYTE   $00            ;TIE BUS FOR CLOCK 8
3023
3024  4A00  01                TI1100: .BYTE   $01            ;TIE BUS FOR CLOCK 0
3025  4A01  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 1
3026  4A02  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 2
3027  4A03  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 3
3028  4A04  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 4
3029  4A05  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 5
3030  4A06  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 6
3031  4A07  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 7
3032  4A08  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 8
3033
3034  4A09  04                TI1101: .BYTE   $04            ;TIE BUS FOR CLOCK 0
3035  4A0A  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 1
3036  4A0B  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 2
3037  4A0C  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 3
3038  4A0D  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 4
3039  4A0E  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 5
3040  4A0F  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 6
3041  4A10  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 7
3042  4A11  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 8
3043
3044  4A12  02                TI1110: .BYTE   $02            ;TIE BUS FOR CLOCK 0
3045  4A13  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 1
3046  4A14  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 2
3047  4A15  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 3
3048  4A16  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 4
3049  4A17  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 5
3050  4A18  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 6
3051  4A19  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 7
3052  4A1A  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 8
3053
3054  4A1B  01                TI1111: .BYTE   $01            ;TIE BUS FOR CLOCK 0
3055  4A1C  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 1
3056  4A1D  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 2
3057  4A1E  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 3
3058  4A1F  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 4
3059  4A20  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 5
3060  4A21  04                        .BYTE   $04            ;TIE BUS FOR CLOCK 6
3061  4A22  02                        .BYTE   $02            ;TIE BUS FOR CLOCK 7
3062  4A23  01                        .BYTE   $01            ;TIE BUS FOR CLOCK 8
3063                          .SBTTL  PROGRAM VARIABLES
```

```
3064
3065  4A24  00                    INDAT:   .BYTE   0          ;DATA BYTE TO ECC
3066  4A25  00                    INDATP:  .BYTE   0          ;DATA BYTE PARITY
3067  4A26  00                    UNITMP:  .BYTE   0          ;UNIT MAP
3068  4A27  00                    SUNIT:   .BYTE   0          ;MB/UNIT SELECT BYTE
3069  4A28  00  00                AMTMSK:  .WORD   0          ;AMTIE MASK WORD
3070  4A2A  00                    DATAA:   .BYTE   0          ;ACTUAL DATA AFTER TRANSLATION
3071  4A2B  00                    DATAAP:  .BYTE   0          ;ACTUAL DATA PARITY AFTER TRANSLATION
3072  4A2C  00                    DATAE:   .BYTE   0          ;EXPECTED DATA AFTER TRANSLATION
3073  4A2D  00                    DATAEP:  .BYTE   0          ;EXPECTED DATA PARITY AFTER TRANSLATION
3074        0000                           .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK   4A28          ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM    4F91          B      =%0000         BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH    4F24          BYTEL    4F23         B4X5    47D2          C      =%0001
CASCT    4F21          CASCTL= 00A0          CASSTA= 00A0          CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1          CBYTH  = 008B          CBYTL  = 008A
CDG1H  = 0087          CDG1L  = 0086         CDG2H  = 0093          CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094         CDVTH  = 008D          CDVTL  = 008C
CHPTIE= 0028           CHOTIE= 0020          CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027           CKAMP    496B  G      CKAM0   48A8  G       CKAM1   48C0  G
CKAM2    48D8  G       CKAM3    48F0  G      CKAM4   4908  G       CKAM5   4920  G
CKAM6    4938  G       CKAM7    4950  G      CKLOP  = 2815          CK5X4   488B
CLEAR    4795          CLECC    4850         CLKCTL= 00F0          CLOCK   4F26
CLRLP    4799          CMC0H  = 0099         CMC0L  = 0098          CMC1H  = 009B
CMC1L  = C09A          CMC2H  = 009D         CMC2L  = 009C          CMC3H  = 009F
CMC3L  = 009E          CMINH  = 0097         CMINL  = 0096          CNTCTL= 00D7
CRCWRD= 0018           CSAVE    4F9D         CSRLH  = 0091          CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084         CTSTH  = 008F          CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080         CXINH  = 0083          CXINL  = 0082
C.     = 0001          C.AVAI= 0080          C.DP   = 0008          C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008         C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001         C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC         C.SER  = 0080          C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002          C4X5    47DB
D      =%0002          DATAA    4A2A         DATAAP  4A2B          DATACT= 00D0
DATAE    4A2C          DATAEP   4A2D         DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8         DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA         DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DIAFLG   4F22         DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045         DONINT= 0010          DSAVE   4F9E
DSE    = 0006          DUMMU    4624  G      DUMMV   46AF  G       DUMMW   4477  G
DUMMX    440B  G       DUMMY    431E  G      D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080         D4X5    47CA          E      =%0003
ECC      4855          ECCBAD= 0042          ECCCHR  4878          ECCCK1  4459  G
ECCCK2   4691  G       ECCCOR= 0019          ECCOK  = 0041          ECCSTA= 001A
ECCTBL   4870          ECCTST= 000E          ECC1    4862          EDATA  = 0095
EOTCLR= 0003           ERFLG    4F93         ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C         ERNUM   4F90          ERRCNT= 00D6
ESAVE    4F9F          EXIT     4792         E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080         E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002          E.UNC  = 0004          E4X5    47F4
FIFORD= 006A           FORMAT   4F25         FOUND   4344          FWDTST= 0061
F4X5     4816          GCRID  = 0089         GCRSET= 0002          GOODTM= 0092
GP4X5    484E          H      =%0004         HLSAVE  4FA0          IE     = 0008
INDAT    4A24          INDATP   4A25         INTSTA= 00E0          ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010          I6.5   = 0020
I7.5     0040          KCALL  = 005F         KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E         KEYBRD= 00C8          KEY1   = 0078
```

```
KEY10 = 006D    KEY11 = 006E    KEY12 = 006F    KEY13 = 005C
KEY14 = 005D    KEY15 = 005E    KEY16 = 005F    KEY17 = 003C
KEY18 = 003D    KEY19 = 003E    KEY2  = 0079    KEY20 = 003F
KEY3  = 007A    KEY4  = 007B    KEY5  = 0074    KEY6  = 0075
KEY7  = 0076    KEY8  = 0077    KEY9  = 006C    KINTA = 006F
KLDAD = 003D    KN0   = 003C    KN1   = 005C    KN2   = 005D
KN3   = 005E    KN4   = 006C    KN5   = 006D    KN6   = 006E
KN7   = 0074    KN8   = 0075    KN9   = 0076    KU2   = 0079
KU3   = 007A    KU8   = 0077    L     =%0005    LBLANK= 000F
LCE   = 000B    LCH   = 000C    LCL   = 000D    LCP   = 000E
LC0   = 0000    LC1   = 0001    LC2   = 0002    LC3   = 0003
LC4   = 0004    LC5   = 0005    LC6   = 0006    LC7   = 0007
LC8   = 0008    LC9   = 0009    LDLEDA= 00CA    LDLEDB= 00CB
LDLEDC= 00CC    LDLEDD= 00CD    LDLEDE= 00CE    LDLEDF= 00CF
LKDIAG= 2800    LKKBD = 004C    LKKEY = 0049    LKLWMG= 0058
LKLWMP= 0055    LKLWPG= 0052    LKLWPP= 004F    LKMOD7= 0046
LKOPR = 0046    LPFLG   4F94    LPNUM   4F92    M     =%0006
MBSEL = 00E0    MB.A  = 0008    MB.B  = 0004    MEMTOP= 4FFF
MINUS = 000A    MM    = 8000    MSE   = 0008    MSGN  = 001A
MTACLR= C000    MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001
MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002
MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001
MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010    MT.Z  = 0008
M.ATA = 0080    M.CAPE= 0020    M.CONT= 0080    M.DEM = 0020
M.EBL = 0010    M.EXC = 0008    M.FAIL= 0008    M.ILR = 0010
M.INIT= 0010    M.OCC = 0020    M.ONLI= 0001    M.PE  = 0040
M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008    M.RUN = 0004
M.SCLK= 0001    M.TRA = 0040    M.UNIT= 0007    M.WCLK= 0040
M.WCLN= 0080    M.WREN= 0080    M5.5  = 0001    M6.5  = 0002
M7.5  = 0004    NOTCAP= 0088    OKAY  = 00FF    OPRRAM= 4300
OPSTRT= 0058    OPVER = 0040    OUT     4971    PADCNT= 00D5
PADCRC= 0080    PDIAG = 0048    PEID  = 008A    PENAB = 004C
PESET = 0001    PL    = 00B1    P04X5   47E1    PRDD  = 004C
PRENF = 009C    PS    = 00B2    PSTAT = 0048    PSW   =%0009
P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080
P.LCS = 0040    P.LWR   0020    P.RDP = 0002    P.RPEN= 0004
P.RPST= 0002    P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020
P.RP3E= 0010    P.SING= 0080    P.STAT= 0002    P.STPE= 0080
P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040
P.WFLP= 0001    P.WPEN= 0010    P.WP0E= 0008    P.WP1E= 0004
P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B
QUEM  = 281E    RAMT  = 0010    RARA  = 0006    RARAI = 0004
RCHBD0= 0048    RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C
RCLRT = 000D    RCMD  = 000B    RCMLP = 0003    RCONT = 0080
RDATA = 0017    RDCLK = 0010    RDON  = 0011    READG = 0007
REND  = 0014    REQST = 2806    RESCHR= 00D1    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 000C    RMCTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
```

```
R.DATA= 0040        R.DON - 0002        R.DRDY= 0010        R.END = 0010
R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008        R.PLOD= 0008
R.PLOO= 0010        R.PLO1= 0020        R.POST= 0020        R.STNM= 0002
R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040
R.VOK = 0080        R00H  = 0081        R00L  = 0080        R01H  = 0083
R01L  = 0082        R02H  = 0085        R02L  = 0084        R03H  = 0087
R03L  = 0086        R04H  = 0089        R04L  = 0088        R05H  = 008B
R05L  = 008A        R06H  = 008D        R06L  = 008C        R07H  = 008F
R07L  = 008E        R10H  = 0091        R10L  = 0090        R11H  = 0093
R11L  = 0092        R12H  = 0095        R12L  = 0094        R13H  = 0097
R13L  = 0096        R14H  = 0099        R14L  = 0098        R15H  = 009B
R15L  = 009A        R16H  = 009D        R16L  = 009C        R17H  = 009F
R17L  = 009E        R7.5  = 0010        SELCLR= 0000        SETATA= 00A1
SID   = 0080        SOD   = 0080        SOE   = 0040        SP    =%0008
SSCLK = 0040        SSTEP = 0005        STACK = 4FFF        STATRM= 4F20
STPCT   4F20        STRSP = 5000        SUNIT   4A27        TAB4X5  4830
TADR00= 0080        TADR01= 0081        TADR02= 0082        TADR03= 0083
TADR04= 0084        TADR05= 0085        TADR06= 0086        TADR07= 0087
TADR10= 0088        TADR11= 0089        TADR12= 008A        TADR13= 008B
TAMT  = C044        TASEL = 0080        TCMD  = 0040        TC.FWD= 0040
TC.INH= 0008        TC.LWR= 0004        TC.REV= 0020        TC.WRT= 0010
TEMP    4F99        TEST1   4300        TEST2   4561        TIETBL  4974
TI0000  4994        TI0001  499D        TI0010  49A6        TI0011  49AF
TI0100  49B8        TI0101  49C1        TI0110  49CA        TI0111  49D3
TI1000  49DC        TI1001  49E5        TI1010  49EE        TI1011  49F7
TI1100  4A00        TI1101  4A09        TI1110  4A12        TI1111  4A1B
TMF   = 0099        TMRDY = 0040        TRANS   4879        TRANS1  487B
TRANS2  4883        TRKENA= 00D2        TRNIN   4840        TRNOUT  4844
TRNTMP  484F        TSET  = 2803        TSTEND= 2818        TSTS  = 0040
TST01A  4414        TST01B  4416        TST01C  436E        TST01F  4389
TST01G  43AF        TST01H  43DA        TST01I  4395        TST01K  4480
TST01L  4496        TST01M  44AE  G     TST01N  44FA        TST01Q  44D9
TST01R  44D0  G     TST01S  44ED        TST01T  44E7  G     TST01U  451B  G
TST01V  453A  G     TST01X  4355        TST01Y  4482        TST02A  462D
TST02B  4647        TST02C  458B        TST02F  45A2        TST02G  45C8
TST02H  45F3        TST02I  45AE        TST02K  46B8        TST02L  46CE
TST02M  46E6  G     TST02N  4732        TST02Q  4711        TST02R  4708  G
TST02S  4725        TST02T  471F  G     TST02U  474F  G     TST02V  476E  G
TST02X  4566        TST02Y  468A        TUSEL0= 00D1        TUSEL1= 00D2
TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002        T.BOT = 0004
T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020
T.PES - 0008        T.PSBJ= 0020        T.PSOJ= 0008        T.PS1J= 0010
T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010        T.SCLK= 0002
T4X5    47C0        UIBG  - 00A1        UNITMP  4A26        UNITSL  4327
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN= 0040        W.ECC - 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV - 0004        W.ROME- 0010        W.RST = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X     =%000A
X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002        X.ROME= 0001
X.WCLK= 0001        Y     %000B         .       4A2E
```

H 6

ERRORS DETECTED: 0

*ECC1.A78/PTP,ECC1=NLIST,PARAM,MACRO,LIST,ECC1
RUN-TIME: 5 7 0 SECONDS
CORE USED: 10K

TABLE OF CONTENTS

```
                        .TITLE  ECC2 - ECC CONTROLLER PART #2
                        .SBTTL  TEST 1 - ECC BAD DATA ONE TRACK NO POINTER
                        ;ID     ECC2-ERROR CORRECTION CONTROLLER PART #2
1332  4300              ST
 (1)                    ; ********************************************************************************
 (1)                    ;*TEST TITLE
 (1)                    ;*---- -----
1333                    ;*ECC BAD DATA ONE TRACK NO POINTER
1334  4300              SD
 (1)                    ; ********************************************************************************
 (1)                    ;*DESCRIPTION
 (1)                    ;*-----------
1335                    ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM SINGLE
1336                    ;*TRACK ERROR CORRECTION WITH THE DATA IN A SINGLE TRACK INCORRECT, AND
1337                    ;*NO POINTERS PROVIDED.  THE PROGRAM CHECKS THAT THE ECC OUTPUT DATA
1338                    ;*IS CORRECTED, AND THAT THE SINGLE TRACK ERROR CORRECT STATUS BIT IS SET.
1339                    ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
1340                    ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
1341                    ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
1342                    ;*FOLLOWING TABLE.
1343                    ;*
1344                    ;*      INPUT/EXPECTED          DATA BIT POSITION       TRACK WITH ECC
1345                    ;*          DATA                WITH ECC CORRECTION       CORRECTION
1346                    ;*      -------------           -------------------     -------------
1347                    ;*
1348                    ;*          XXXX0000                    0                       2
1349                    ;*          XXXX0001                    1                       8
1350                    ;*          XXXX0010                    2                       1
1351                    ;*          XXXX0011                    3                       9
1352                    ;*          XXXX0100                    4                       3
1353                    ;*          XXXX0101                    5                       5
1354                    ;*          XXXX0110                    6                       6
1355                    ;*          XXXX0111                    7                       7
1356                    ;*          XXXX1000                    P                       4
1357                    ;*          XXXX1001                    1                       8
1358                    ;*          XXXX1010                    2                       1
1359                    ;*          XXXX1011                    3                       9
1360                    ;*          XXXX1100                    4                       3
1361                    ;*          XXXX1101                    5                       5
1362                    ;*          XXXX1110                    6                       6
1363                    ;*          XXXX1111                    7                       7
1364                    ;*
1365                    ;*      X = DON'T CARE BITS
1366  4300              SP
 (1)                    ; ********************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
1367                    ;*BGNTST
1368                    ;*      SET NORMAL READ PATH CLOCK
1369                    ;*      CALL SUBROUTINE CLEAR
1370                    ;*      SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1371                    ;*      CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1372                    ;*      CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1373                    ;*      CLEAR THE TIE BUS
```

```
                        ;*    CLEAR THE INPUT DATA
                        ;*    CLEAR BAD TRANSLATION POINTERS
                        ;*  BGNDO
                        ;*  : ENABLE THE READ PATH CLOCK
                        ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                        ;*  : CLOCK THE FIFO'S
                        ;*  : ISSUE CLEAR ALL COMMAND
                        ;*  : WAIT
                        ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                        ;*  : CALL SUBROUTINE CLECC
                        ;*  : SET THE LOOP COUNT TO 7
                        ;*  : BGNDO
                        ;*  : :  GET THE INPUT DATA BYTE
                        ;*  : :  CALL SUBROUTINE ECC
                        ;*  : :  DECREMENT THE LOOP COUNT
                        ;*  : :  DO UNTIL THE LOOP COUNT=0
                        ;*  : ENDDO
                        ;*  : SET THE LOOP COUNT TO 3
                        ;*  : BGNDO
                        ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
                        ;*  : :  CALCULATE INCORRECT TRANSLATION TRACK
                        ;*  : :  STORE POINTER FOR SUBROUTINE
                        ;*  : :  CALL SUBROUTINE TRK
                        ;*  : :  CALL SUBROUTINE BADT4X5
                        ;*  : :  INIT THE LOOP COUNT TO 5
                        ;*  : :  BGNDO
                        ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                        ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                        ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                        ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                        ;*  : :  :  DECREMENT THE LOOP COUNT
                        ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                        ;*  : :  :  DO UNTIL LOOP COUNT = 0
                        ;*  : :  ENDDO
                        ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
                        ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
                        ;*  : :  CALCULATE INCORRECT TRANSLATION TRACK
                        ;*  : :  STORE POINTER FOR SUBROUTINE
                        ;*  : :  CALL SUBROUTINE TRK
                        ;*  : :  CALL SUBROUTINE BADT4X5
                        ;*  : :  INIT THE LOOP COUNT TO 5
                        ;*  : :  BGNDO
                        ;*  : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
                        ;*  : :  :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
                        ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                        ;*  : :  :  DECREMENT THE LOOP COUNT
                        ;*  : :  :  DO UNTIL THE LOOP COUNT = 0
                        ;*  : :  ENDDO
                        ;*  : :  DECREMENT THE LOOP COUNT
                        ;*  : :  DO UNTIL THE LOOP COUNT = 0
                        ;*  : ENDDO
                        ;*  : ISSUE DIAGNOSTIC READ COMMAND
                        ;*  : SET UP WATCHDOG TIMER COUNT
                        ;*  : BGNDO
```

```
1428                                    ;*  : :    SINGLE STEP THE READ PATH
1429                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1430                                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1431                                    ;*  : ENDDO
1432                                    ;*  : IF WATCHDOG TIMER=0
1433                                    ;*  : :    THEN-ERROR
1434                                    ;*  : :    ELSE-CONTINUE
1435                                    ;*  : ENDIF
1436                                    ;*  : INIT THE LOOP COUNT TO 5
1437                                    ;*  : BGNDO
1438                                    ;*  : :    SINGLE STEP THE READ PATH
1439                                    ;*  : :    DECREMENT THE LOOP COUNT
1440                                    ;*  : :    DO UNTIL LOOP COUNT = 0
1441                                    ;*  : ENDDO
1442                                    ;*  : SET UP WATCHDOG TIMER COUNT
1443                                    ;*  : BGNDO
1444                                    ;*  : :  SINGLE STEP THE READ PATH
1445                                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1446                                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1447                                    ;*  : ENDDO
1448                                    ;*  : IF WATCHDOG TIMER=0
1449                                    ;*  : :  THEN-ERROR-EXIT TEST
1450                                    ;*  : :  ELSE-CONTINUE
1451                                    ;*  : ENDIF
1452                                    ;*  : CALL SUBROUTINE CKDONE
1453                                    ;*  : SINGLE STEP THE READ PATH
1454                                    ;*  : INIT THE  LOOP COUNT TO 7
1455                                    ;*  : BGNDO
1456                                    ;*  : :  SINGLE STEP THE READ PATH
1457                                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1458                                    ;*  : :  IF NOT EQUAL
1459                                    ;*  : :  :  THEN-ECC ERROR
1460                                    ;*  : :  :  ELSE-CONTINUE
1461                                    ;*  : :  ENDIF
1462                                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1463                                    ;*  : :  IF NOT EQUAL
1464                                    ;*  : :  :  THEN-ECC ERROR
1465                                    ;*  : :  :  ELSE-CONTINUE
1466                                    ;*  : :  ENDIF
1467                                    ;*  : :  DECREMENT THE LOOP COUNT
1468                                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
1469                                    ;*  : ENDDO
1470                                    ;*  : SINGLE STEP THE READ PATH
1471                                    ;*  : IF EXPECTED NOT EQUAL
1472                                    ;*  : :  THEN-ECC ERROR
1473                                    ;*  : :  ELSE-CONTINUE
1474                                    ;*  : ENDIF
1475                                    ;*  : INPUT THE ECC STATUS
1476                                    ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
1477                                    ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
1478                                    ;*  : :  THEN-CONTINUE
1479                                    ;*  : :  ELSE-ERROR
1480                                    ;*  : ENDIF
1481                                    ;*  : INCREMENT THE INPUT DATA
```

```
                                        ;*  : DO UNTIL INPUT DATA = 0.
                                        ;*  ENDDO
                                        ;*ENDTST
 4300                                   SE
  (1)                                   ; ***************************************************************************
  (1)                                   ;*ERRORS
  (1)                                   ;*------
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 01
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 02
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                        ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*FATAL ERROR - MICRO TEST ABORTED
                                        ;*
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 03
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                                        ;*SENT TO THE ECC
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*FATAL ERROR - MICRO TEST ABORTED
                                        ;*
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 04
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA FROM THE ECC-INCORRECT
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 05
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT - ZERO SHOULD = ONE
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 06
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                        ;*BYTE/SCLK COUNT NUMBER - LLL
```

```
                                        ;*
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 07
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC2 MICRO TEST 01
                                        ;*ECC2 MICRO ERROR 10
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
 (1)                                    ; ************************************************************************************

                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
4300  3E  01              7.0                       MVI     A,@1          ;DEFINE THE TEST NUMBER
4302  CD  03  28         18.0                       CALL    TSET          ;SETUP THE TEST
                                        ;%ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                        ;&M8951, M8950'S

4305                                            REQ     @7,0,0,0,0
4305  CD  06  28         18.0                       CALL    REQST
4308  00                                          .BYTE   0             ;DATA PATTERN NUMBER
4309  00  00                                      .WORD   0             ;SYSTEM "0" COUNT
430B  00  00                                      .WORD   0             ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
430D  00                                          .BYTE   0             ;DATA COMPARE FLAG IF =1
430E  07                                          .BYTE   @7            ;REQUEST CODE
430F                                            RIN     R12L
430F  DB  94               10.0                 IN      R12L          ;READ R12L INTO AC
4311  7F                    4.0                 MOV     A,A           ;RETRY LINK
4312  32  B2  48           13.0         STA     UNITMP
4315  A7                    4.0         ANA     A
4316  C2  27  43           10.0         JNZ     UNITSL
4319                                    ERR     EXIT,DUMMY
                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
4319  CD  09  28           18.0                 CALL    ERLP          ;PROCESS ERROR - DO 2.3
      0001                              MSGN    =       MSGN+1                ;UPDATE MESSAGE NUMBER FOR THIS
431C  01                                .BYTE   MSCN          ;MESSAGE NUMBER ID
431D  00                                .BYTE
431E  CD  15  28           18.0         DUMMY:: CALL    CKLOP         ;CHECK LOOP FUNCTION - DO 2.3
4321  DA  36  47           10.0                 JC      EXIT          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;>FATAL ERROR - TEST ABORTED
4324  C3  36  47           10.0         JMP     EXIT

4327  06  00                7.0  UNITSL: MVI     B,0
4329  3A  B2  48           13.0         LDA     UNITMP           ;DID THE USER SPECIFY TU
432C  E6  01                7.0         ANI     @01              ;PORT 0?
432E  C2  44  43           10.0         JNZ     FOUND            ;YES-GO USE PORT #0
```

```
1569  4331  04              4.0          INR     B               ;NO-UPDATE POINTER TO PORT #1
1570  4332  3A  B2  48     13.0          LDA     UNITMP          ;DID THE USER SPECIFY TU
1571  4335  E6  02           7.0          ANI     @02             ;PORT 1?
1572  4337  C2  44  43     10.0          JNZ     FOUND           ;YES-GO USE PORT #1
1573  433A  04              4.0          INR     B               ;NO-UPDATE POINTER TO PORT #2
1574  433B  3A  B2  48     13.0          LDA     UNITMP          ;DID THE USER SPECIFY TU
1575  433E  E6  04           7.0          ANI     @04             ;PORT 2?
1576  4340  C2  44  43     10.0          JNZ     FOUND           ;YES-GO USE PORT #2
1577  4343  04              4.0          INR     B               ;NO-ASSUME PORT #3
1578  4344  CD  39  47     18.0  FOUND:  CALL    CLEAR
1579  4347  DB  E0         10.0          IN      INTSTA
1580  4349  E6  80          7.0          ANI     BIT7
1581  434B  B0              4.0          ORA     B
1582  434C  D3  E0         10.0          OUT     MBSEL
1583  434E  32  B3  48     13.0          STA     SUNIT
1584
1585  4351  3E  10          7.0  TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1586  4353  D3  F0         10.0          OUT     CLKCTL
1587  4355  3E  10          7.0          MVI     A,W.GCR
1588  4357  D3  D3         10.0          OUT     WMCCTL
1589
1590  4359  3E  04          7.0          MVI     A,P.RPEN
1591  435B  D3  4C         10.0          OUT     PENAB
1592
1593  435D  3E  60          7.0          MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1594  435F  D3  48         10.0          OUT     PDIAG           ;LCS, LWR AND DPEN
1595  4361  AF              4.0          XRA     A
1596  4362  D3  44         10.0          OUT     TAMT
1597  4364  D3  D2         10.0          OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1598  4366  D3  D2         10.0          OUT     TRKENA          ;THE TRANSLATOR
1599  4368  D3  0A         10.0          OUT     RTIEB           ;CLEAR THE TIE BUS
1600  436A  32  B0  48     13.0          STA     INDAT
1601  436D  32  4B  48     13.0          STA     BAD1
1602  4370  32  4C  48     13.0          STA     BAD2
1603  4373  32  4D  48     13.0          STA     BAD3
1604  4376                               ROUT    R05H
 (1)  4376  D3  8B         10.0                  OUT     R05H            ;WRITE AC INTO R05H
 (1)  4378  7F              4.0                  MOV     A,A             ;RETRY LINK
1605
1606  4379  3E  A8          7.0  TST01C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1607                                                                     ;SET TIE BUS JAM AND PLO DISABLE
1608  437B  D3  09         10.0          OUT     RPCTL
1609  437D  D3  08         10.0          OUT     RFIFOL          ;CLOCK THE FIFO'S
1610  437F  3E  0D          7.0          MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1611  4381  D3  0B         10.0          OUT     RCMD
1612
1613  4383  00              4.0          NOP
1614  4384  00              4.0          NOP                     ;WAIT
1615  4385  00              4.0          NOP
1616  4386  00              4.0          NOP
1617  4387  00              4.0          NOP
1618
1619  4388  3E  A9          7.0          MVI     A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1620                                                                     ;SET TIE BUS JAM AND PLO DISABLE
```

```
1621  438A  D3  09      10.0          OUT   RPCTL
1622  438C  CD  4E  48   18.0          CALL  CLECC          ;CLEAR THE ECC CHARACTER
1623  438F  16  07       7.0          MVI   D,@7           ;SET UP THE LOOP COUNT
1624  4391  3A  B0  48   13.0  TST01F: LDA   INDAT          ;GET THE INPUT DATA
1625  4394  CD  53  48   18.0          CALL  ECC            ;CALCULATE ECC FOR
1626  4397  15           4.0          DCR   D              ;THE 7 INPUT BYTES
1627  4398  C2  91  43   10.0          JNZ   TST01F
1628
1629  439B  0E  03       7.0          MVI   C,@3           ;INIT THE LOOP COUNT
1630  439D  3A  B0  48   13.0  TST01I: LDA   INDAT          ;GET THE INPUT DATA
1631  43A0                       ROUT  R05L
 (1)  43A0  D3  8A      10.0                OUT   R05L             ;WRITE AC INTO R05L
 (1)  43A2  7F          4.0                 MOV   A,A              ;RETRY LINK
1632  43A3  32  3B  48   13.0          STA   TRNIN          ;FILL THE TRANSLATOR
1633  43A6  32  3C  48   13.0          STA   TRNIN+1        ;SUBROUTINE INPUT
1634  43A9  32  3D  48   13.0          STA   TRNIN+2        ;BUFFER
1635  43AC  32  3E  48   13.0          STA   TRNIN+3        ;
1636  43AF  CD  89  48   18.0          CALL  TRK
1637  43B2  CD  64  47   18.0          CALL  BADT4X5        ;TRANSLATE THE SUBGROUP
1638
1639  43B5  21  3F  48   10.0          LXI   H,TRNOUT       ;GET POINTER TO TRANSLATE DATA TABLE
1640  43B8  06  05       7.0          MVI   B,@5           ;SET UP LOOP COUNT
1641  43BA  7E           7.0  TST01G: MOV   A,M           ;GET A DATA BYTE
1642  43BB  D3  40      10.0          OUT   TCMD           ;STORE DATA IN COMMAND ADDRESS
1643  43BD  23           6.0          INX   H              ;POINT TO DATA PARITY
1644  43BE  7E           7.0          MOV   A,M            ;GET THE DATA PARITY
1645
1646  43BF  07           4.0          RLC                  ;POSITION FOR OUTPUT
1647  43C0  F6  60       7.0          ORI   P.LWR!P.LCS    ;OR IN CONTROL BITS
1648  43C2  D3  48      10.0          OUT   PDIAG          ;OUTPUT THE DATA PARITY
1649
1650  43C4  D3  08      10.0          OUT   RFIFOL         ;CLOCK DATA INTO THE FIFO'S
1651
1652  43C6  23           6.0          INX   H              ;UPDATE THE TABLE POINTER
1653  43C7  05           4.0          DCR   B              ;DECREMENT LOOP COUNT
1654  43C8  C2  BA  43   10.0          JNZ   TST01G         ;DO UNTIL LOOP COUNT = 0
1655
1656  43CB  3A  B0  48   13.0          LDA   INDAT          ;GET THE INPUT DATA
1657  43CE  32  3B  48   13.0          STA   TRNIN          ;FILL THE FIRST THREE
1658  43D1  32  3C  48   13.0          STA   TRNIN+1        ;BYTES OF THE TRANSLATOR
1659  43D4  32  3D  48   13.0          STA   TRNIN+2        ;SUBROUTINE WITH THE INPUT DATA
1660  43D7  3A  76  48   13.0          LDA   ECCCHR         ;STORE THE CALCULATED
1661  43DA  32  3E  48   13.0          STA   TRNIN+3        ;ECC CHARACTER AS THE LAST CHARACTER
1662  43DD  CD  89  48   18.0          CALL  TRK
1663
1664  43E0  CD  64  47   18.0          CALL  BADT4X5        ;TRANSLATE THE SECOND SUBGROUP
1665
1666  43E3  21  3F  48   10.0          LXI   H,TRNOUT       ;GET THE POINTER TO INPUT DATA TABLE
1667  43E6  06  05       7.0          MVI   B,@5           ;SET UP THE LOOP COUNT
1668. 43E8  7E           7.0  TST01H: MOV   A,M           ;GET A DATA BYTE
1669  43E9  D3  40      10.0          OUT   TCMD           ;STORE DATA IN CMD ADDRESS
1670  43EB  23           6.0          INX   H              ;POINT TO DATA PARITY BIT
1671  43EC  7E           7.0          MOV   A,M            ;GET THE DATA PARITY
1672  43ED  07           4.0          RLC                  ;POSITION FOR OUTPUT
```

```
1673  43EE  F6  60        7.0          ORI   P.LWR.P.LCS      ;OR IN CONTROL BITS
1674  43F0  D3  48       10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
1675  43F2  D3  08       10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFOS
1676  43F4  23            6.0          INX   H                ;UPDATE THE TABLE POINTER
1677  43F5  05            4.0          DCR   B                ;DECREMENT THE LOOP COUNT
1678  43F6  C2  E8  43   10.0          JNZ   TST01H           ;DO UNTIL LOOP COUNT=0
1679
1680  43F9  0D            4.0          DCR   C                ;PUT THE GROUP IN
1681  43FA  C2  9D  43   10.0          JNZ   TST01I           ;THE FIFO TWICE
1682
1683  43FD  3E  0B        7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
1684  43FF  D3  0B       10.0          OUT   RCMD             ;COMMAND
1685  4401  11  01  00   10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
1686  4404  21  A8  FD   10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
1687  4407  D3  0C       10.0  2$:     OUT   RINST            ;STEP THE READ PATH
1688  4409  DB  01       10.0          IN    RPCHI            ;DATA READY SET?
1689  440B  E6  10        7.0          ANI   R.DRDY
1690  440D  C2  22  44   10.0          JNZ   TST01A           ;YES-GO PROCESS
1691  4410  19           10.0          DAD   D                ;WATCH DOG TIMEOUT?
1692  4411  D2  07  44   10.0          JNC   2$               ;NO-CONTINUE
1693  4414                             ERR   TST01C,DUMMX,1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4414  CD  09  28   18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0002                            MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4417  02                              .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4418  01                              .BYTE 1
 (1)  4419  CD  15  28   18.0          DUMMX:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  441C  DA  79  43   10.0                JC    TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1694                               ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1695                               ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1696                               ;<FATAL ERROR - MICRO TEST ABORTED
1697  441F  C3  36  47   10.0          JMP   EXIT
1698
1699  4422  D3  0C       10.0  TST01A: OUT   RINST            ;SINGLE STEP THE CLOCK
1700  4424  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1701  4426  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1702  4428  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1703  442A  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1704  442C  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1705  442E  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1706  4430  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1707  4432  D3  0C       10.0          OUT   RINST            ;SINGLE STEP THE CLOCK
1708  4434  11  01  00   10.0          LXI   D,1              ;SET WATCHDOG TIMER INCREMENT
1709  4437  21  A8  FD   10.0          LXI   H,-600           ;SET WATCHDOG COUNT TO 600
1710  443A  D3  0C       10.0  3$:     OUT   RINST            ;SINGLE STEP THE READ PATH
1711  443C  DB  01       10.0          IN    RPCHI            ;DATA READY SET?
1712  443E  E6  10        7.0          ANI   R.DRDY
1713  4440  C2  55  44   10.0          JNZ   TST01K           ;YES-GO PROCESS
1714  4443  19           10.0          DAD   D                ;NO-WATCHDOG TIMEOUT?
1715  4444  D2  3A  44   10.0          JNC   3$               ;NO-CONTINUE
```

```
1717  4447                                          ERR     TST01C,DUMMW,1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4447  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0003                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  444A  03                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  444B  01                                            .BYTE   1
 (1)  444C  CD  15  28        18.0                DUMMW:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.5
 (1)  444F  DA  79  43        10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1718                                    ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONF GROUP HAS BEEN
1719                                    ;>SENT TO THE ECC
1720                                    ;<FATAL ERROR - MICRO TEST ABORTED
1721  4452  C3  36  47        10.0                JMP     EXIT
1722
1723  4455  16  01              7.0   TST01K: MVI     D,@1            ;INIT THE LOOP COUNT
1724  4457  AF                    4.0   TST01Y: XRA     A
1725  4458                                    ROUT    R05H
 (1)  4458  D3  8B             10.0                    OUT     R05H            ;WRITE AC INTO R05H
 (1)  445A  7F                  4.0                    MOV     A,A             ;RETRY LINK
1726  445B  3A  B0  48         13.0                LDA     INDAT           ;GET THE INPUT DATA
1727  445E                                    ROUT    EDATA
 (1)  445E  D3  95             10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4460  7F                  4.0                    MOV     A,A             ;RETRY LINK
1728  4461  A7                    4.0                ANA     A               ;SET CONDITION BITS
1729  4462  3E  00              7.0                MVI     A,0             ;CLEAR THE ACC.
1730  4464  E2  68  44         10.0                JPO     1$              ;ODD # OF ONES - CONTINUE
1731  4467  3C                    4.0                INR     A               ;EVEN # OF ONES - SET PARITY
1732  4468  32  B1  48         13.0   1$:          STA     INDATP          ;STORE THE PARITY BIT
1733  446B  D3  0C             10.0   TST01L: OUT     RINST           ;SINGLE STEP THE READ LOGIC
1734  446D  3A  B0  48         13.0                LDA     INDAT
1735  4470  47                    4.0                MOV     B,A
1736  4471  DB  19             10.0                IN      ECCCOR
1737  4473  B8                    4.0                CMP     B
1738  4474  CA  83  44         10.0                JZ      TST01M
1739  4477                                    ROUT    ADATA
 (1)  4477  D3  94             10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4479  7F                  4.0                    MOV     A,A             ;RETRY LINK
1740  447A  78                    4.0            MOV     A,B             ;GET THE BYTE COUNT
1741  447B                                    ROUT    R05L            ;STORE IN CAS
 (1)  447B  D3  8A             10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  447D  7F                  4.0                    MOV     A,A             ;RETRY LINK
1742  447E                                    ERRB    TST01C,TST01M,@1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  447E  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4481  04                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4482  01                                            .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  4483  CD  15  28        18.0                TST01M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4486  DA  79  43        10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1743                                    ;>CORRECTED DATA FROM THE ECC-INCORRECT
1744  4489  DB  15             10.0                IN      RPSTA           ;GET THE ECC PARITY BIT
1745  448B  E6  80              7.0                ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
1746  448D  07                    4.0                RLC                     ;POSITION BIT FOR COMPARE
1747  448E  4F                    4.0                MOV     C,A             ;SAVE IN REG. C
1748  448F  3A  B1  48         13.0                LDA     INDATP          ;GET THE EXPECTED PARITY BIT
```

```
1749  4492  A7                    4.0          ANA     A                ;SET THE CONDITION BITS
1750  4493  CA  AE  44            10.0          JZ      TST01QO          ;GO CHECK FOR PARITY=0
1751  4496  79                    4.0          MOV     A,C              ;ELSE CHECK FOR PARITY=1
1752  4497  FE  01                7.0          CPI     @1
1753  4499  CA  A5  44            10.0          JZ      TST01RG
1754  449C  78                    4.0          MOV     A,B
1755  449D                                      ROUT    R05L
 (1)  449D  D3  8A                10.0                  OUT     R05L             ;WRITE AC INTO R05L
 (1)  449F  7F                    4.0                   MOV     A,A              ;RETRY LINK
1756  44A0                                      ERR     TST01C,TST01RG,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44A0  CD  09  28            18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0005                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A3  05                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44A4  01                                          .BYTE   @1
 (1)  44A5  CD  15  28            18.0          TST01RG::       CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A8  DA  79  43            10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1757                                      ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1758                                      ;<
1759  44AB  C3  C2  44            10.0          JMP     TST01SZ          ;CONTINUE WITH TEST
1760  44AE  79                    4.0  TST01QO: MOV     A,C              ;CHECK FOR PARITY = ZERO
1761  44AF  A7                    4.0          ANA     A                ;SET CONDITION BITS
1762  44B0  CA  BC  44            10.0          JZ      TST01TP          ;CONTINUE IF ZERO
1763  44B3  78                    4.0          MOV     A,B
1764  44B4                                      ROUT    R05L
 (1)  44B4  D3  8A                10.0                  OUT     R05L             ;WRITE AC INTO R05L
 (1)  44B6  7F                    4.0                   MOV     A,A              ;RETRY LINK
1765  44B7                                      ERR     TST01C,TST01TP,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44B7  CD  09  28            18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0006                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44BA  06                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44BB  01                                          .BYTE   @1
 (1)  44BC  CD  15  28            18.0          TST01TP::       CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44BF  DA  79  43            10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1766                                      ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1767                                      ;<
1768
1769  44C2  7A                    4.0  TST01SZ: MOV     A,D              ;GET THE CHARACTER COUNT
1770  44C3  3C                    4.0          INR     A                ;INCREMENT
1771  44C4  FE  08                7.0          CPI     $8               ;DONE?
1772  44C6  CA  CD  44            10.0          JZ      TST01N           ;YES-GO CHECK ECC
1773  44C9  57                    4.0          MOV     D,A              ;NO-CONTINUE
1774  44CA  C3  57  44            10.0          JMP     TST01Y
1775
1776  44CD  D3  0C                10.0  TST01N: OUT     RINST            ;SINGLE STEP THE READ PATH
1777  44CF  3A  76  48            13.0          LDA     ECCCHR           ;GET THE EXPECTED ECC CHARACTER
1778  44D2                                      ROUT    EDATA            ;SAVE IN CAS
 (1)  44D2  D3  95                10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  44D4  7F                    4.0                   MOV     A,A              ;RETRY LINK
1779  44D5  47                    4.0          MOV     B,A              ;SAVE IN REGISTER B
1780  44D6  DB  19                10.0          IN      ECCCOR
1781  44D8                                      ROUT    ADATA
 (1)  44D8  D3  94                10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
```

```
 (1)  44DA  7F                    4.0                 MOV     A,A             ;RETRY LINK
1782  44DB  B8                    4.0         CMP     B
1783  44DC  CA  EA  44            10.0        JZ      TST01U
1784  44DF  3A  B0  48            13.0        LDA     INDAT           ;GET THE DATA/COUNT
1785  44E2                                     ROUT    R05L            ;WRITE TO THE CAS
 (1)  44E2  D3  8A                10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  44E4  7F                    4.0                 MOV     A,A             ;RETRY LINK
1786  44E5                                     ERR     TST01C,TST01U,1
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E5  CD  09  28            18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E8  07                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E9  01                               .BYTE   1
 (1)  44EA  CD  15  28            18.0        TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44ED  DA  79  43            10.0                JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1787                                     ;>CORRECTED ECC CHARACTER INCORRECT
1788                                     ;<
1789  44F0  DB  1A                10.0        IN      ECCSTA          ;GET THE ECC STATUS
1790  44F2  E6  0F                 7.0        ANI     @17             ;REMOVE UNWANTED BITS
1791  44F4  FE  09                 7.0        CPI     E.STEC!E.PNTR
1792  44F6  CA  0C  45             10.0       JZ      TST01V
1793  44F9                                     ROUT    ADATA
 (1)  44F9  D3  94                10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44FB  7F                    4.0                 MOV     A,A             ;RETRY LINK
1794  44FC  3E  09                 7.0        MVI     A,E.STEC!E.PNTR
1795  44FE                                     ROUT    EDATA
 (1)  44FE  D3  95                10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4500  7F                    4.0                 MOV     A,A             ;RETRY LINK
1796  4501  3A  B0  48            13.0        LDA     INDAT           ;GET THE DATA/COUNT
1797  4504                                     ROUT    R05L            ;WRITE TO THE CAS
 (1)  4504  D3  8A                10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  4506  7F                    4.0                 MOV     A,A             ;RETRY LINK
1798  4507                                     ERRB    TST01C,TST01V,1
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4507  CD  12  28            18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  450A  08                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450B  01                               .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  450C  CD  15  28            18.0        TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  450F  DA  79  43            10.0                JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1799                                     ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1800                                     ;<
1801  4512  3A  B0  48            13.0        LDA     INDAT
1802  4515  3C                     4.0        INR     A
1803  4516  32  B0  48            13.0        STA     INDAT
1804  4519  C2  79  43            10.0        JNZ     TST01C
1805  451C                                     ENDTST  TST01X
 (1)                                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  451C                                   REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  451C  CD  06  28            18.0                CALL    REQST
 (2)  451F  00                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  4520  00  00                           .WORD                   ;SYSTEM '*' COUNT
 (2)  4522  00  00                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4524  00                               .BYTE                   ;DATA COMPARE FLAG IF =1
```

```
                                                                .BYTE   7               ;REQUEST CODE
  (2)  4525  07                                                                        
  (1)  4526  3A  9A  4F      13.0          LDA     ITERA         ;GET ITERATION COUNT
  (1)  4529  3D                4.0         DCR     A             ;DOWNCOUNT
  (1)  452A  32  9A  4F      13.0          STA     ITERA         ;SAVE COUNT
  (1)  452D  F2  51  43      10.0          JP      TST01X        ;DO TEST UNTIL TILL = 0
1806   4530  CD  39  47      18.0          CALL    CLEAR         ;CLEAR ALL TU PORTS
```

```
                                .SBTTL  TEST 2 - ECC BAD DATA ONE TRACK GOOD POINTER
4533                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ECC BAD DATA ONE TRACK GOOD POINTER
4533                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM SINGLE
                                ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN A SINGLE TRACK, AND A
                                ;*CORRECT POINTER ALSO PROVIDED.  THE PROGRAM CHECKS THAT THE DATA IS
                                ;*CORRECTED, AND THAT THE SINGLE TRACK ERROR CORRECT STATUS BIT IS SET.
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITION       TRACK WITH ECC
                                ;*          DATA                WITH ECC CORRECTION       CORRECTION
                                ;*      --------------          -------------------     --------------
                                ;*
                                ;*          XXXX0000                    0                     2
                                ;*          XXXX0001                    1                     8
                                ;*          XXXX0010                    2                     1
                                ;*          XXXX0011                    3                     9
                                ;*          XXXX0100                    4                     3
                                ;*          XXXX0101                    5                     5
                                ;*          XXXX0110                    6                     6
                                ;*          XXXX0111                    7                     7
                                ;*          XXXX1000                    P                     4
                                ;*          XXXX1001                    1                     8
                                ;*          XXXX1010                    2                     1
                                ;*          XXXX1011                    3                     9
                                ;*          XXXX1100                    4                     3
                                ;*          XXXX1101                    5                     5
                                ;*          XXXX1110                    6                     6
                                ;*          XXXX1111                    7                     7
                                ;*
                                ;*      X = DON'T CARE BITS
                                ;*
                                ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                                ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED
                                ;*          DATA                        POINTER GENERATED
                                ;*      --------------                  -----------------
                                ;*
                                ;*          XX00XXXX                    HISTORY
                                ;*          XX01XXXX                    PHASE TIE
                                ;*          XX10XXXX                    AMTIE
                                ;*          XX11XXXX                    ILLEGAL 5X4
                                ;*
```

```
                                ;*      X = DON'T CARE BITS
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*    CLEAR THE INPUT DATA
                                ;*    CLEAR BAD TRANSLATION POINTERS
                                ;*  BGNDO
                                ;*  : ENABLE THE READ PATH CLOCK
                                ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  : CLOCK THE FIFO'S
                                ;*  : ISSUE CLEAR ALL COMMAND
                                ;*  : WAIT
                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                                ;*  : CALL SUBROUTINE CLECC
                                ;*  : SET THE LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  GET THE INPUT DATA BYTE
                                ;*  : :  CALL SUBROUTINE ECC
                                ;*  : :  DECREMENT THE LOOP COUNT
                                ;*  : :  DO UNTIL THE LOOP COUNT=0
                                ;*  : ENDDO
                                ;*  : SET THE LOOP COUNT TO 3
                                ;*  : BGNDO
                                ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
                                ;*  : :  CALL SUBROUTINE TRK
                                ;*  : :  CALL SUBROUTINE BADT4X5
                                ;*  : :  INIT THE LOOP COUNT TO 5
                                ;*  : :  BGNDO
                                ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                                ;*  : :  :  DECREMENT THE LOOP COUNT
                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*  : :  :  DO UNTIL LOOP COUNT = 0
                                ;*  : :  ENDDO
                                ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
                                ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
                                ;*  : :  CALL SUBROUTINE TRK
                                ;*  : :  CALL SUBROUTINE BADT4X5
                                ;*  : :  INIT THE LOOP COUNT TO 5
                                ;*  : :  BGNDO
                                ;*  : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
                                ;*  : :  :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                                ;*  : :  :  DECREMENT THE LOOP COUNT
                                ;*  : :  :  DO UNTIL THE LOOP COUNT = 0
```

```
1908                                    ;*  : :  ENDDO
1909                                    ;*  : :  DECREMENT THE LOOP COUNT
1910                                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
1911                                    ;*  : ENDDO
1912                                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1913                                    ;*  : SET UP WATCHDOG TIMER COUNT
1914                                    ;*  : BGNDO
1915                                    ;*  : :    SINGLE STEP THE READ PATH
1916                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1917                                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1918                                    ;*  : ENDDO
1919                                    ;*  : IF WATCHDOG TIMER=0
1920                                    ;*  : :    THEN-ERROR
1921                                    ;*  : :    ELSE-CONTINUE
1922                                    ;*  : ENDIF
1923                                    ;*  : GET THE TIE BUS TABLE ADDRESS
1924                                    ;*  : SINGLE STEP THE READ PATH
1925                                    ;*  : INIT THE LOOP COUNT TO 0
1926                                    ;*  : BGNDO
1927                                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT
1928                                    ;*  : :    :  THEN-CALL SUBROUTINE POINTER
1929                                    ;*  : :    :  ELSE-CLEAR TIE BUS POINTER
1930                                    ;*  : :    ENDIF
1931                                    ;*  : :    LOAD THE TIE BUS REGISTER
1932                                    ;*  : :    SINGLE STEP THE READ PATH
1933                                    ;*  : :    DECREMENT THE LOOP COUNT
1934                                    ;*  : :    DO UNTIL LOOP COUNT = 9
1935                                    ;*  : ENDDO
1936                                    ;*  : SET UP WATCHDOG TIMER COUNT
1937                                    ;*  : BGNDO
1938                                    ;*  : :  SINGLE STEP THE READ PATH
1939                                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1940                                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1941                                    ;*  : ENDDO
1942                                    ;*  : IF WATCHDOG TIMER=0
1943                                    ;*  : :  THEN-ERROR-EXIT TEST
1944                                    ;*  : :  ELSE-CONTINUE
1945                                    ;*  : ENDIF
1946                                    ;*  : SINGLE STEP THE READ PATH
1947                                    ;*  : INIT THE  LOOP COUNT TO 7
1948                                    ;*  : BGNDO
1949                                    ;*  : :  SINGLE STEP THE READ PATH
1950                                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1951                                    ;*  : :  IF NOT EQUAL
1952                                    ;*  : :  :  THEN-ECC ERROR
1953                                    ;*  : :  :  ELSE-CONTINUE
1954                                    ;*  : :  ENDIF
1955                                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1956                                    ;*  : :  IF NOT EQUAL
1957                                    ;*  : :  :  THEN-ECC ERROR
1958                                    ;*  : :  :  ELSE-CONTINUE
1959                                    ;*  : :  ENDIF
1960                                    ;*  : :  DECREMENT THE LOOP COUNT
1961                                    ;*  : :  DO UNTIL THE LOOP COUNT - 0
```

```
1962                                ;*   : ENDDO
1963                                ;*   : SINGLE STEP THE READ PATH
1964                                ;*   : IF EXPECTED NOT EQUAL
1965                                ;*   : :  THEN-ECC ERROR
1966                                ;*   : :  ELSE-CONTINUE
1967                                ;*   : ENDIF
1968                                ;*   : INPUT THE ECC STATUS
1969                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
1970                                ;*   : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT
1971                                ;*   : :  THEN-CONTINUE
1972                                ;*   : :  ELSE-ERROR
1973                                ;*   : ENDIF
1974                                ;*   : INCREMENT THE INPUT DATA
1975                                ;*   : DO UNTIL INPUT DATA = 0
1976                                ;*  ENDDO
1977                                ;*ENDTST
1978  4533                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1979                                ;*ECC2 MICRO TEST 02
1980                                ;*ECC2 MICRO ERROR 11
1981                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
1982                                ;*M8951, M8950'S
1983                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1984                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1985                                ;*FATAL ERROR - MICRO TEST ABORTED
1986                                ;*
1987                                ;*ECC2 MICRO TEST 02
1988                                ;*ECC2 MICRO ERROR 12
1989                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
1990                                ;*M8951, M8950'S
1991                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1992                                ;*SENT TO THE ECC
1993                                ;*FATAL ERROR - MICRO TEST ABORTED
1994                                ;*
1995                                ;*ECC2 MICRO TEST 02
1996                                ;*ECC2 MICRO ERROR 13
1997                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
1998                                ;*M8951, M8950'S
1999                                ;*CORRECTED DATA FROM THE ECC-INCORRECT
2000                                ;*BYTE/SCLK COUNT NUMBER = LLL
2001                                ;*ACTUAL = NNNN
2002                                ;*EXPECTED = NNNN
2003                                ;*
2004                                ;*ECC2 MICRO TEST 02
2005                                ;*ECC2 MICRO ERROR 14
2006                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
2007                                ;*M8951, M8950'S
2008                                ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2009                                ;*BYTE/SCLK COUNT NUMBER = LLL
2010                                ;*
2011                                ;*ECC2 MICRO TEST 02
2012                                ;*ECC2 MICRO ERROR 15
```

```
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC2 MICRO TEST 02
                                        ;*ECC2 MICRO ERROR 16
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC2 MICRO TEST 02
                                        ;*ECC2 MICRO ERROR 17
                                        ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4533                                    S
 (1)                                    ; ***************************************************************************
                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1) 4533  3E  02            7.0                 MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1) 4535  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
                                        ;&M8951, M8950'S
                                                REQ     @7,0,0,0,0
 (1) 4538  CD  06  28       18.0                         CALL    REQST
 (1) 4538  00                                            .BYTE   0               ;DATA PATTERN NUMBER
 (1) 453C  00  00                                        .WORD   0               ;SYSTEM "0" COUNT
 (1) 453E  00  00                                        .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 4540  00                                            .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1) 4541  07                                            .BYTE   @7              ;REQUEST CODE
                                        
4542  3E  10            7.0  TST02X:  MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
4544  D3  F0           10.0          OUT     CLKCTL
4546  3E  10            7.0          MVI     A,W.GCR
4548  D3  D3           10.0          OUT     WMCCTL
                                        
454A  CD  39  47       18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
454D  3A  B3  48       13.0          LDA     SUNIT
4550  D3  E0           10.0          OUT     MBSEL
                                        
4552  3E  04            7.0          MVI     A,P.RPEN        ;ENABLE THE READ PATH
4554  D3  4C           10.0          OUT     PENAB
                                        
4556  3E  60            7.0          MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
4558  D3  48           10.0          OUT     PDIAG           ;LCS, LWR
455A  AF                4.0          XRA     A
455B  D3  44           10.0          OUT     TAMT
```

```
2058  455D  D3  D2        10.0          OUT   TRKENA            ;CLEAR ALL TRACKS FROM
2059  455F  D3  0A        10.0          OUT   RTIEB             ;CLEAR THE TIE BUS REGISTER
2060  4561  D3  D2        10.0          OUT   TRKENA
2061  4563  32  4B  48    13.0          STA   BAD1              ;CLEAR THE INVALID
2062  4566  32  4C  48    13.0          STA   BAD2              ;TRACK TRANSLATIONS
2063  4569  32  4D  48    13.0          STA   BAD3              ;POINTERS
2064  456C  32  B0  48    13.0          STA   INDAT             ;CLEAR THE INPUT DATA BYTE
2065
2066  456F  3E  A8         7.0  TST02C: MVI   A,R.PLO1!R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2067                                                          ;SET TIE BUS JAM AND PLO DISABLE
2068  4571  D3  09        10.0          OUT   RPCTL
2069  4573  D3  08        10.0          OUT   RFIFOL            ;CLOCK THE FIFO'S
2070  4575  3E  0D         7.0          MVI   A,RCLRT           ;ISSUE CLEAR ALL COMMAND
2071  4577  D3  0B        10.0          OUT   RCMD
2072
2073  4579  00            4.0          NOP                     ;WAIT
2074  457A  00            4.0          NOP
2075  457B  00            4.0          NOP
2076  457C  00            4.0          NOP
2077  457D  00            4.0          NOP
2078
2079  457E  3E  A9         7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2080                                                          ;SET TIE BUS JAM AND PLO DISABLE
2081  4580  D3  09        10.0          OUT   RPCTL
2082  4582  CD  4E  48    18.0          CALL  CLECC             ;CLEAR THE ECC CHARACTER
2083  4585  16  07         7.0          MVI   D,@7              ;SET UP THE LOOP COUNT
2084  4587  3A  B0  48    13.0  TST02F: LDA   INDAT             ;GET THE INPUT DATA
2085  458A  CD  53  48    18.0          CALL  ECC               ;CALCULATE ECC FOR
2086  458D  15            4.0          DCR   D                 ;THE 7 INPUT BYTES
2087  458E  C2  87  45    10.0          JNZ   TST02F
2088
2089  4591  0E  03         7.0          MVI   C,@3              ;INIT THE LOOP COUNT
2090  4593  3A  B0  48    13.0  TST02I: LDA   INDAT             ;GET THE INPUT DATA
2091  4596  32  3B  48    13.0          STA   TRNIN             ;FILL THE TRANSLATOR
2092  4599  32  3C  48    13.0          STA   TRNIN+1           ;SUBROUTINE INPUT
2093  459C  32  3D  48    13.0          STA   TRNIN+2           ;BUFFER
2094  459F  32  3E  48    13.0          STA   TRNIN+3           ;
2095  45A2  CD  89  48    18.0          CALL  TRK
2096  45A5  CD  64  47    18.0          CALL  BADT4X5           ;TRANSLATE THE SUBGROUP
2097
2098  45A8  21  3F  48    10.0          LXI   H,TRNOUT          ;GET POINTER TO TRANSLATE DATA TABLE
2099  45AB  06  05         7.0          MVI   B,@5              ;SET UP LOOP COUNT
2100  45AD  7E             7.0  TST02G: MOV   A,M              ;GET A DATA BYTE
2101  45AE  D3  40        10.0          OUT   TCMD              ;STORE DATA IN COMMAND ADDRESS
2102  45B0  23            6.0          INX   H                 ;POINT TO DATA PARITY
2103  45B1  7E             7.0          MOV   A,M               ;GET THE DATA PARITY
2104
2105  45B2  07            4.0          RLC                     ;POSITION FOR OUTPUT
2106  45B3  F6  60         7.0          ORI   P.LWR!P.LCS       ;OR IN CONTROL BITS
2107  45B5  D3  48        10.0          OUT   PDIAG             ;OUTPUT THE DATA PARITY
2108
2109  45B7  D3  08        10.0          OUT   RFIFOL            ;CLOCK DATA INTO THE FIFO'S
2110
2111  45B9  23            6.0          INX   H                 ;UPDATE THE TABLE POINTER
```

```
2112  45BA  05              4.0          DCR   B              ;DECREMENT LOOP COUNT
2113  45BB  C2  AD  45      10.0          JNZ   TST02G         ;DO UNTIL LOOP COUNT = 0
2114
2115  45BE  3A  B0  48      13.0          LDA   INDAT          ;GET THE INPUT DATA
2116  45C1  32  3B  48      13.0          STA   TRNIN          ;FILL THE FIRST THREE
2117  45C4  32  3C  48      13.0          STA   TRNIN+1        ;BYTES OF THE TRANSLATOR
2118  45C7  32  3D  48      13.0          STA   TRNIN+2        ;SUBROUTINE WITH THE INPUT DATA
2119  45CA  3A  76  48      13.0          LDA   ECCCHR         ;STORE THE CALCULATED
2120  45CD  32  3E  48      13.0          STA   TRNIN+3        ;ECC CHARACTER AS THE LAST CHARACTER
2121  45D0  CD  89  48      18.0          CALL  TRK
2122
2123  45D3  CD  64  47      18.0          CALL  BADT4X5        ;TRANSLATE THE SECOND SUBGROUP
2124
2125  45D6  21  3F  48      10.0          LXI   H,TRNOUT       ;GET THE POINTER TO INPUT DATA TABLE
2126  45D9  06  05           7.0          MVI   B,@5           ;SET UP THE LOOP COUNT
2127  45DB  7E               7.0  TST02H: MOV   A,M           ;GET A DATA BYTE
2128  45DC  D3  40          10.0          OUT   TCMD           ;STORE DATA IN CMD ADDRESS
2129  45DE  23               6.0          INX   H              ;POINT TO DATA PARITY BIT
2130  45DF  7E               7.0          MOV   A,M            ;GET THE DATA PARITY
2131  45E0  07               4.0          RLC                  ;POSITION FOR OUTPUT
2132  45E1  F6  60           7.0          ORI   P.LWR!P.LCS    ;OR IN CONTROL BITS
2133  45E3  D3  48          10.0          OUT   PDIAG          ;OUTPUT THE DATA PARITY
2134  45E5  D3  08          10.0          OUT   RFIFOL         ;CLOCK DATA INTO THE FIFOS
2135  45E7  23               6.0          INX   H              ;UPDATE THE TABLE POINTER
2136  45E8  05               4.0          DCR   B              ;DECREMENT THE LOOP COUNT
2137  45E9  C2  DB  45      10.0          JNZ   TST02H         ;DO UNTIL LOOP COUNT=0
2138
2139  45EC  0D               4.0          DCR   C              ;PUT THE GROUP IN
2140  45ED  C2  93  45      10.0          JNZ   TST02I         ;THE FIFO TWICE
2141
2142  45F0  3E  0B           7.0          MVI   A,DIARD        ;LOAD THE DIAGNOSTIC READ
2143  45F2  D3  0B          10.0          OUT   RCMD           ;COMMAND
2144  45F4  11  01  00      10.0          LXI   D,1            ;SET WATCH DOG INCREMENT
2145  45F7  21  A8  FD      10.0          LXI   H,-600         ;SET WATCH DOG COUNT TO 600
2146  45FA  D3  0C          10.0  2$:     OUT   RINST          ;STEP THE READ PATH
2147  45FC  DB  01          10.0          IN    RPCHI          ;DATA REA Y SET?
2148  45FE  E6  10           7.0          ANI   R.DRDY
2149  4600  C2  15  46      10.0          JNZ   TST02A         ;YES-GO PROCESS
2150  4603  19               10.0          DAD   D              ;WATCH DOG TIMEOUT?
2151  4604  D2  FA  45      10.0          JNC   2$             ;NO-CONTINUE
2152  4607                                ERR   TST02C,DUMMU
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4607  CD  09  28      18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0009                         MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460A  09                          .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  460B  00                          .BYTE
 (1)  460C  CD  15  28      18.0  DUMMU:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  460F  DA  6F  45      10.0          JC    TST02C           ;LOOP ADDRESS IF LOOP SPECIFIED
2153                            ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2154                            ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2155                            ;<FATAL ERROR - MICRO TEST ABORTED
2156  4612  C3  36  47      10.0          JMP   EXIT
2157
2158  4615  D3  0C          10.0  TST02A: OUT   RINST
```

```
2159  4617  06 00          7.0          MVI   B,$0            ;INIT THE BIT TIME COUNT (LOOP COUNT)
2160  4619  00             4.0  TST02B: NOP
2161  461A  3A AF 48       13.0          LDA   TIETIM          ;GET THE TIME FOR TIE BUS JAMING
2162  461D  B8             4.0          CMP   B               ;IS IT TIME TO JAM THE TIE BUS?
2163  461E  C2 27 46       10.0          JNZ   4$              ;NO - CONTINUE
2164  4621  CD 0F 48       18.0          CALL  POINTER         ;YES - GO GET A REAL POINTER
2165  4624  C3 2A 46       10.0          JMP   5$              ;
2166  4627  AF             4.0  4$:     XRA   A               ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
2167  4628  D3 0A          10.0          OUT   RTIEB           ;LOAD THE POINTER
2168  462A  D3 0C          10.0  5$:     OUT   RINST           ;STEP CLOCK
2169  462C  04             4.0          INR   B               ;DECREMENT THE LOOP COUNT
2170  462D  78             4.0          MOV   A,B
2171  462E  FE 09          7.0          CPI   $9              ;DONE
2172  4630  C2 19 46       10.0          JNZ   TST02B          ;DO UNTIL LOOP COUNT = ZERO
2173  4633  AF             4.0          XRA   A
2174  4634  D3 0A          10.0          OUT   RTIEB           ;CLEAR THE TIE BUS REGISTER
2175  4636  D3 0C          10.0          OUT   RINST           ;CLOCK THE READ PATH
2176  4638  D3 0C          10.0          OUT   RINST           ;CLOCK THE READ PATH
2177  463A  11 01 00       10.0          LXI   D,1             ;SET WATCHDOG TIMER INCREMENT
2178  463D  21 A8 FD       10.0          LXI   H,-600          ;SET WATCHDOG COUNT TO 600
2179  4640  D3 0C          10.0  3$:     OUT   RINST           ;SINGLE STEP THE READ PATH
2180  4642  DB 01          10.0          IN    RPCHI           ;DATA READY SET?
2181  4644  E6 10          7.0           ANI   R.DRDY
2182  4646  C2 5B 46       10.0          JNZ   TST02K          ;YES-GO PROCESS
2183  4649  19             10.0          DAD   D               ;NO-WATCHDOG TIMEOUT?
2184  464A  D2 40 46       10.0          JNC   3$              ;NO-CONTINUE
2185  464D                               ERR   TST02C,DUMMV
                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)
 (1)  464D  CD 09 28       18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                              MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4650  0A                                .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4651  00                                .BYTE
 (1)  4652  CD 15 28       18.0  DUMMV::       CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4655  DA 6F 45       10.0                JC    TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2186                               ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2187                               ;>SENT TO THE ECC
2188                               ;<FATAL ERROR - MICRO TEST ABORTED
2189  4658  C3 36 47       10.0          JMP   EXIT
2190
2191  465B  16 01          7.0  TST02K: MVI   D,@1            ;INIT THE LOOP COUNT
2192  465D  AF             4.0  TST02Y: XRA   A
2193  465E                             ROUT  R05H
 (1)  465E  D3 8B          10.0                OUT   R05H            ;WRITE AC INTO R05H
 (1)  4660  7F             4.0                 MOV   A,A             ;RETRY LINK
2194  4661  3A B0 48       13.0          LDA   INDAT           ;GET THE INPUT DATA
2195  4664                             ROUT  EDATA
 (1)  4664  D3 95          10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  4666  7F             4.0                 MOV   A,A             ;RETRY LINK
2196  4667  A7             4.0          ANA   A               ;SET CONDITION BITS
2197  4668  3E 00          7.0          MVI   A,0             ;CLEAR THE ACC.
2198  466A  E2 6E 46       10.0          JPO   1$              ;ODD # OF ONES - CONTINUE
2199  466D  3C             4.0          INR   A               ;EVEN # OF ONES - SET PARITY
2200  466E  32 B1 48       13.0  1$:     STA   INDATP          ;STORE THE PARITY BIT
2201  4671  D3 0C          10.0  TST02L: OUT   RINST           ;SINGLE STEP THE READ LOGIC
```

```
2202  4673  3A  B0  48   13.0          LDA     INDAT
2203  4676  47               4.0       MOV     B,A
2204  4677  DB  19          10.0       IN      ECCCOR
2205  4679  B8               4.0       CMP     B
2206  467A  CA  89  46       10.0       JZ      TST02M
2207  467D                             ROUT    ADATA
 (1)  467D  D3  94           10.0               OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  467F  7F               4.0                MOV     A,A             ;RETRY LINK
2208  4680  78               4.0       MOV     A,B             ;GET THE BYTE COUNT
2209  4681                             ROUT    R05L            ;STORE IN CAS
 (1)  4681  D3  8A           10.0               OUT     R05L            ;WRITE AC INTO R05L
 (1)  4683  7F               4.0                MOV     A,A             ;RETRY LINK
2210  4684                             ERRB    TST02C,TST02M,@1
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4684  CD  12  28   18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4687  08                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4688  01                                .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  4689  CD  15  28   18.0           TST02M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  468C  DA  6F  45   10.0                    JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2211                             ;>CORRECTED DATA FROM THE ECC-INCORRECT
2212  468F  DB  15          10.0       IN      RPSTA           ;GET THE ECC PARITY BIT
2213  4691  E6  80           7.0       ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2214  4693  07               4.0       RLC                     ;POSITION BIT FOR COMPARE
2215  4694  4F               4.0       MOV     C,A             ;SAVE IN REG. C
2216  4695  3A  B1  48       13.0       LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2217  4698  A7               4.0       ANA     A               ;SET THE CONDITION BITS
2218  4699  CA  B4  46       10.0       JZ      TST02QO         ;GO CHECK FOR PARITY=0
2219  469C  79               4.0       MOV     A,C             ;ELSE CHECK FOR PARITY=1
2220  469D  FE  01           7.0       CPI     @1
2221  469F  CA  AB  46       10.0       JZ      TST02RG
2222  46A2  78               4.0       MOV     A,B
2223  46A3                             ROUT    R05L
 (1)  46A3  D3  8A           10.0               OUT     R05L            ;WRITE AC INTO R05L
 (1)  46A5  7F               4.0                MOV     A,A             ;RETRY LINK
2224  46A6                             ERR     TST02C,TST02RG,@1
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46A6  CD  09  28   18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A9  0C                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46AA  01                                .BYTE   @1
 (1)  46AB  CD  15  28   18.0           TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46AE  DA  6F  45   10.0                    JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2225                             ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2226                             ;<
2227  46B1  C3  C8  46       10.0       JMP     TST02SZ         ;CONTINUE WITH TEST
2228  46B4  79               4.0  TST02QO: MOV  A,C             ;CHECK FOR PARITY = ZERO
2229  46B5  A7               4.0       ANA     A               ;SET CONDITION BITS
2230  46B6  CA  C2  46       10.0       JZ      TST02TP         ;CONTINUE IF ZERO
2231  46B9  78               4.0       MOV     A,B
2232  46BA                             ROUT    R05L
 (1)  46BA  D3  8A           10.0               OUT     R05L            ;WRITE AC INTO R05L
 (1)  46BC  7F               4.0                MOV     A,A             ;RETRY LINK
```

```
2234  46BD                                        ERR     TST02C,TST02TP,@1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46BD  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000D                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46C0  0D                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46C1  01                                           .BYTE   @1
 (1)  46C2  CD  15  28        18.0               TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46C5  DA  6F  45        10.0                       JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2235                                    ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2236                                    ;<
2237  46C8  7A                   4.0   TST02SZ: MOV     A,D             ;GET THE CHARACTER COUNT
2238  46C9  3C                   4.0            INR     A               ;INCREMENT
2239  46CA  FE  08               7.0            CPI     $8              ;DONE?
2240  46CC  CA  D3  46          10.0            JZ      TST02N          ;YES-GO CHECK ECC
2241  46CF  57                   4.0            MOV     D,A             ;NO-CONTINUE
2242  46D0  C3  5D  46          10.0            JMP     TST02Y
2243
2244  46D3  D3  0C              10.0   TST02N:  OUT     RINST           ;SINGLE STEP THE READ PATH
2245  46D5  3A  76  48          13.0            LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
2246  46D8                                    ROUT    EDATA           ;SAVE IN CAS
 (1)  46D8  D3  95              10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46DA  7F                   4.0                    MOV     A,A             ;RETRY LINK
2247  46DB  47                   4.0            MOV     B,A             ;SAVE IN REGISTER B
2248  46DC  DB  19              10.0            IN      ECCCOR
2249  46DE                                    ROUT    ADATA
 (1)  46DE  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46E0  7F                   4.0                    MOV     A,A             ;RETRY LINK
2250  46E1  B8                   4.0            CMP     B
2251  46E2  CA  F0  46          10.0            JZ      TST02U
2252  46E5  3A  B0  48          13.0            LDA     INDAT           ;GET THE DATA/COUNT
2253  46E8                                    ROUT    R05L            ;WRITE TO THE CAS
 (1)  46E8  D3  8A              10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  46EA  7F                   4.0                    MOV     A,A             ;RETRY LINK
2254  46EB                                    ERR     TST02C,TST02U,1
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46EB  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000E                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46EE  0E                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46EF  01                                           .BYTE   1
 (1)  46F0  CD  15  28        18.0               TST02U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46F3  DA  6F  45        10.0                       JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2255                                    ;>CORRECTED ECC CHARACTER INCORRECT
2256                                    ;<
2257  46F6  DB  1A              10.0            IN      ECCSTA          ;GET THE ECC STATUS
2258  46F8  E6  0F               7.0            ANI     @17             ;REMOVE UNWANTED BITS
2259  46FA  FE  01               7.0            CPI     E.STEC
2260  46FC  CA  12  47          10.0            JZ      TST02V
2261  46FF                                    ROUT    ADATA
 (1)  46FF  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4701  7F                   4.0                    MOV     A,A             ;RETRY LINK
2262  4702  3E  01               7.0            MVI     A,E.STEC
2263  4704                                    ROUT    EDATA
 (1)  4704  D3  95              10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4706  7F                   4.0                    MOV     A,A             ;RETRY LINK
```

```
2264  4707  3A  B0  48   13.0          LDA     INDAT           ;GET THE DATA/COUNT
2265  470A                             ROUT    R05L            ;WRITE TO THE CAS
 (1)  470A  D3  8A       10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  470C  7F            4.0                    MOV     A,A             ;RETRY LINK
2266  470D                             ERRB    TST02C,TST02V,1
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  470D  CD  12  28   18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000F                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4710  0F                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4711  01                                    .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4712  CD  15  28   18.0          TST02V::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4715  DA  6F  45   10.0                    JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2267                             ;>ECC STATUS INCORRECT AFTER A DATA GROUP
2268                             ;<
2269  4718  3A  B0  48   13.0          LDA     INDAT
2270  471B  3C            4.0          INR     A
2271  471C  32  B0  48   13.0          STA     INDAT
2272  471F  C2  6F  45   10.0          JNZ     TST02C
2273  4722                             ENDTST  TST02X
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4722                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4722  CD  06  28   18.0                    CALL    REQST
 (2)  4725  C0                                   .BYTE                   ;DATA PATTERN NUMBER
 (2)  4726  00  00                               .WORD                   ;SYSTEM '*' COUNT
 (2)  4728  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  472A  00                                   .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  472B  07                                   .BYTE   7               ;REQUEST CODE
 (1)  472C  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  472F  3D            4.0          DCR     A               ;DOWNCOUNT
 (1)  4730  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  4733  F2  42  45   10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
2274  4736  C3  18  28   10.0  EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*****************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*****************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*****************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*****************************************************************************
                              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                              12.0          PUSH    B
                               7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
                              10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
                               7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
                              10.0          OUT     MBSEL           ;RESET TO THIS PORT
                               7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                              10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                               4.0          XRA     A               ;CLEAR TU COMMAND A
                              10.0          OUT     TCMD            ;
                               7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                              10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                               7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
                              10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
                               4.0          XRA     A
                              10.0          OUT     TAMT            ;CLEAR AMTIE WORD
                              10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                              10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
                               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                               4.0          MOV     A,B
                               7.0          CPI     4               ;DONE?
                              10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                              10.0          POP     B               ;RESET B & C
                              10.0          POP     PSW             ;ALL DONE
                              10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
                                        SSUB
                                        ;*****************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*4X5 TRANSLATE A SUBGROUP
                                        SD
                                        ;*****************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                        ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE, AT THE ENTRY
                                        ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                        ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                        ;*BUT NOT CORRECT FOR THE INPUT DATA.
                                        SP
                                        ;*****************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    CLEAR THE BAD TRANSLATION POINTERS
                                        ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                        ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                        ;*    SET LOOP COUNT TO 1
                                        ;*    BGNDO
                                        ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                        ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                        ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                        ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                        ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                        ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                        ;*    :        TO BUILD THE INDEX)
                                        ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                        ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                        ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                        ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                        ;*    :        DATA BYTE.
                                        ;*    :    INCREMENT THE LOOP COUNT
                                        ;*    :    DO UNTIL THE LOOP COUNT = 10
                                        ;*    ENDDO
                                        ;*ENDSUB
                                        S
                                        ;*****************************************************************************

4764  F5              12.0  BADT4X5: PUSH   PSW             ;SAVE REGISTER A + PSW
4765  C3  73  47       10.0         JMP    BAD
4768  F5              12.0  T4X5:    PUSH   PSW             ;SAVE REGISTER A + PSW
4769  AF               4.0         XRA    A               ;CLEAR A
476A  32  4B  48       13.0         STA    BAD1            ;CLEAR BAD TRANSLATION 1
476D  32  4C  48       13.0         STA    BAD2            ;CLEAR BAD TRANSLATION 2
4770  32  4D  48       13.0         STA    BAD3            ;CLEAR BAD TRANSLATION 3
4773  C5              12.0  BAD:     PUSH   B               ;SAVE REGISTER B + C
4774  D5              12.0         PUSH   D               ;SAVE REGISTER D + E
4775  E5              12.0         PUSH   H               ;SAVE REGISTER H + L
```

```
2364                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
2365                                        ;CLEAR THE TRANSLATED DATA TABLE.
2366  4776  06  0A          7.0           MVI     B,10            ;SET UP LOOP COUNT
2367  4778  11  3F  48     10.0           LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2368  477B  AF               4.0           XRA     A               ;CLEAR A
2369  477C  12               7.0  D4X5:    STAX    D               ;STORE IN THE TABLE
2370  477D  13               6.0           INX     D               ;UPDATE TABLE POINTER
2371  477E  05               4.0           DCR     B               ;DECREMENT LOOP COUNT
2372  477F  C2  7C  47      10.0           JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
2373  4782  0E  01           7.0           MVI     C,1             ;SET UP TRACK COUNT
2374  4784  06  04           7.0  B4X5:    MVI     B,4             ;SET UP BIT COUNT
2375  4786  11  3B  48     10.0           LXI     D,TRNIN
2376  4789  AF               4.0           XRA     A               ;CLEAR THE GROUP POSITION COUNT
2377  478A  32  49  48     13.0           STA     GP4X5
2378  478D  1A               7.0  C4X5:    LDAX    D               ;GET A DATA BYTE
2379  478E  A7               4.0           ANA     A               ;SET CONDITION BITS
2380  478F  E2  93  47      10.0           JPO     PO4X5           ;ODD PARITY LEAVE CARRY CLEAR
2381  4792  37               4.0           STC                     ;EVEN PARITY SET CARRY
2382  4793  1F               4.0  PO4X5:   RAR                     ;SHIFT OUT DESIRED BIT
2383  4794  12               7.0           STAX    D               ;STORE RESULT BACK IN TEMP TABLE
2384  4795  3A  49  48     13.0           LDA     GP4X5           ;GET THE GROUP POSITION BYTE
2385  4798  17               4.0           RAL                     ;PUT IN THIS DATA BIT
2386  4799  32  49  48     13.0           STA     GP4X5           ;SAVE THE UPDATED GROUP POSITION BYTE
2387  479C  13               6.0           INX     D               ;UPDATE THE TABLE POINTER
2388  479D  05               4.0           DCR     B               ;DECREMENT THE BIT COUNT
2389  479E  C2  8D  47      10.0           JNZ     C4X5            ;DO UNTIL ALL 4 BITS ARE RECEIVED
2390
2391                                        ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2392
2393                                        ;PERFORM A DOUBLE PRECISION SHIFT RIGHT ON THE TRANSLATED DATA
2394  47A1  06  05           7.0           MVI     B,5             ;SET UP LOOP COUNT
2395  47A3  11  3F  48     10.0           LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2396  47A6  13               6.0  E4X5:    INX     D               ;POINT TO PARITY BIT
2397  47A7  1A               7.0           LDAX    D               ;GET PARITY BIT
2398  47A8  A7               4.0           ANA     A               ;CLEAR THE CARRY BIT
2399  47A9  1F               4.0           RAR                     ;SHIFT TO THE CARRY BIT
2400  47AA  12               7.0           STAX    D               ;STORE IT BACK
2401  47AB  1B               6.0           DCX     D               ;DECREMENT TO DATA BITS
2402  47AC  1A               7.0           LDAX    D               ;GET THE DATA BITS
2403  47AD  1F               4.0           RAR                     ;SHIFT IN CARRY BITS
2404  47AE  12               7.0           STAX    D               ;STORE BACK IN TABLE
2405  47AF  13               6.0           INX     D               ;POINT TO NEXT TABLE ENTRY
2406  47B0  13               6.0           INX     D               ;
2407  47B1  05               4.0           DCR     B               ;DECREMENT LOOP COUNT
2408  47B2  C2  A6  47      10.0           JNZ     E4X5            ;DO UNTIL LOOP COUNT=0
2409  47B5  3A  4B  48     13.0           LDA     BAD1            ;IS A BAD POINTER SPECIFIED?
2410  47B8  A7               4.0           ANA     A               ;SET CONDITION BITS
2411  47B9  CA  C0  47      10.0           JZ      BAD1C           ;NO-CONTINUE
2412  47BC  B9               4.0           CMP     C               ;YES-IS IT THIS DATA POSITION?
2413  47BD  CA  D6  47      10.0           JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2414  47C0  3A  4C  48     13.0  BAD1C:   LDA     BAD2            ;IS A BAD POINTER SPECIFIED
2415  47C3  A7               4.0           ANA     A               ;SET CONDITION BITS
2416  47C4  CA  CB  47      10.0           JZ      BAD2C           ;NO-CONTINUE
2417  47C7  B9               4.0           CMP     C               ;YES-IS IT THIS DATA POSITION
```

```
2418  47C8  CA  D6  47   10.0          JZ    INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2419  47CB  3A  4D  48   13.0  BAD2C:  LDA   BAD3            ;IS A BAD POINTER SPECIFIED
2420  47CE  A7               4.0          ANA   A             ;SET CONDITION BITS
2421  47CF  CA  DD  47   10.0          JZ    BAD3C           ;NO-PERFORM PROPER TRANSLATION
2422  47D2  B9               4.0          CMP   C             ;YES-IS IT THIS DATA POSITION
2423  47D3  C2  DD  47   10.0          JNZ   BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
2424  47D6  3A  49  48   13.0  INV4X5: LDA   GP4X5           ;GET THE GROUP POSITION COUNT
2425  47D9  3C               4.0          INR   A             ;INCREMENT-TO INCORRECT TRANSLATION
2426  47DA  32  49  48   13.0          STA   GP4X5           ;STORE IT
2427  47DD  11  2B  48   10.0  BAD3C:  LXI   D,TAB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
2428  47E0  26  00            7.0          MVI   H,0           ;CLEAR REGISTER H
2429  47E2  3A  49  48   13.0          LDA   GP4X5           ;GET GROUP POSITION COUNT
2430  47E5  E6  0F            7.0          ANI   $0F
2431  47E7  6F               4.0          MOV   L,A
2432  47E8  19              10.0          DAD   D             ;ADD GROUP POSITION COUNT TO TABLE
2433  47E9  7E               7.0          MOV   A,M           ;GET TRANSLATED DATA
2434  47EA  32  4A  48   13.0          STA   TRNTMP
2435                                ;''TRNTMP'' = THE TRANSLATED DATA
2436
2437  47ED  11  3F  48   10.0          LXI   D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2438  47F0  06  05            7.0          MVI   B,5           ;SET UP LOOP COUNT
2439  47F2  3A  4A  48   13.0  F4X5:   LDA   TRNTMP          ;GET TRANSLATED DATA
2440  47F5  13               6.0          INX   D             ;POINT TO PARITY POSITION
2441  47F6  A7               4.0          ANA   A             ;CLEAR CARRY BIT
2442  47F7  1F               4.0          RAR                 ;SHIFT OUT A BIT
2443  47F8  32  4A  48   13.0          STA   TRNTMP          ;STORE TRANSLATED DATA
2444  47FB  1A               7.0          LDAX  D             ;GET THE PARITY BIT (ALWAYS ZERO)
2445  47FC  17               4.0          RAL                 ;ROLL IN THE CARRY BIT
2446  47FD  12               7.0          STAX  D             ;STORE AS THE PARITY BIT
2447  47FE  13               6.0          INX   D             ;UPDATE TABLE POINTER
2448  47FF  05               4.0          DCR   B             ;DECREMENT THE LOOP COUNT
2449  4800  C2  F2  47   10.0          JNZ   F4X5            ;DO UNTIL LOOP COUNT=0
2450
2451                                ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2452
2453  4803  0C               4.0          INR   C             ;DECREMENT THE TRACK COUNT
2454  4804  79               4.0          MOV   A,C
2455  4805  FE  0A            7.0          CPI   10
2456  4807  C2  84  47   10.0          JNZ   B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2457  480A  E1              10.0          POP   H             ;RESTORE REGISTER H + L
2458  480B  D1              10.0          POP   D             ;RESTORE REGISTER D + E
2459  480C  C1              10.0          POP   B             ;RESTORE REGISTER B + C
2460  480D  F1              10.0          POP   PSW           ;RESTORE REGISTER A + PSW
2461  480E  C9              10.0          RET                 ;RETURN TO USER
2462
```

```
                                        .SBTTL  SUBROUTINE CALCULATE POINTER
                                        ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                                        ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                        ;*
                                        ;*      INPUT/EXPECTED
                                        ;*          DATA                        POINTER GENERATED
                                        ;*      --------------                  -----------------
                                        ;*
                                        ;*         XX00XXXX                     HISTORY
                                        ;*         XX01XXXX                     PHASE TIE
                                        ;*         XX10XXXX                     AMTIE
                                        ;*         XX11XXXX                     ILLEGAL 5X4
                                        ;*
                                        ;*      X = DON'T CARE BITS
480F                                    SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        
480F  F5              12.0  POINTER: PUSH   PSW             ;SAVE THE ACCUMULATOR
4810  C5              12.0           PUSH   B               ;SAVE B + C
4811  3A  B0  48      13.0           LDA    INDAT           ;GET INPUT DATA
4814  E6  30           7.0           ANI    $30             ;REMOVE THE WORKING BITS
4816  07               4.0           RLC                    ;POSITION THE DATA
4817  07               4.0           RLC
4818  07               4.0           RLC
4819  07               4.0           RLC
481A  21  27  48      10.0           LXI    H,POINTBL       ;GET BASE ADDRESS OF TABLE
481D  4F               4.0           MOV    C,A
481E  06  00           7.0           MVI    B,0
4820  09              10.0           DAD    B
4821  7E               7.0           MOV    A,M             ;GET THE ECC POINTER
4822  D3  0A          10.0           OUT    RTIEB           ;SET THE TIE BUS VALUE
4824  C1              10.0           POP    B               ;RESTORE B + C
4825  F1              10.0           POP    PSW             ;RESTORE ACCUMULATOR
4826  C9              10.0           RET
                                        
4827  01                       POINTBL: .BYTE  1
4828  02                                .BYTE  2
4829  04                                .BYTE  4
482A  08                                .BYTE  8
```

```
                                .SBTTL  TABLE 4 X 5 TRANSLATION

                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
                                ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
                                ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
                                ;
                                ;       0000                              10011
                                ;       0001                              11011
                                ;       0010                              01001
                                ;       0011                              11001
                                ;       0100                              10111
                                ;       0101                              10101
                                ;       0110                              01101
                                ;       0111                              11101
                                ;       1000                              01011
                                ;       1001                              10010
                                ;       1010                              01010
                                ;       1011                              11010
                                ;       1100                              01111
                                ;       1101                              10110
                                ;       1110                              01110
                                ;       1111                              11110
                                
482B  13                        TAB4X5: .BYTE   @23
482C  1B                                .BYTE   @33
482D  09                                .BYTE   @11
482E  19                                .BYTE   @31
482F  17                                .BYTE   @27
4830  15                                .BYTE   @25
4831  0D                                .BYTE   @15
4832  1D                                .BYTE   @35
4833  0B                                .BYTE   @13
4834  12                                .BYTE   @22
4835  0A                                .BYTE   @12
4836  1A                                .BYTE   @32
4837  0F                                .BYTE   @17
4838  16                                .BYTE   @26
4839  0E                                .BYTE   @16
483A  1E                                .BYTE   @36
```

```
                                        .SBTTL  SUBROUTINE VARIABLES

                        TRNIN:  .BLKB   4       ;TABLE CONTAINING THE 4 DATA BYTES TO
                                                ;BE TRANSLATED
                        TRNOUT: .BLKB   10      ;TRANSLATED DATA TABLE DATA AFTER 4X5
                                                ;TRANSLATION
                        GP4X5:  .BYTE   0       ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
                                                ;BE TRANSLATED
                        TRNTMP: .BYTE   0       ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
                        BAD1:   .BYTE   0       ;BAD TRANSLATION POINTER 1
                        BAD2:   .BYTE   0       ;BAD TRANSLATION POINTER 2
                        BAD3:   .BYTE   0       ;BAD TRANSLATION POINTER 3
```

```
                                        .SBTTL  SUBROUTINE CLEAR ECC
                                        S
                                        ; ***************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION ''ECCCHR'' -
                                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
                                        S
                                        ; ***************************************************************************

484E  AF                      4.0  CLECC:  XRA     A              ;CLEAR THE ACCUMULATOR
484F  32   76   48           13.0         STA     ECCCHR         ;CLEAR THE ECC CHARACTER
4852  C9                     10.0         RET                    ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        S
                                        ; ***************************************************************************
 (1)                                    ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        S
 (1)                                    ; ***************************************************************************

4853  21  76  48      10.0  ECC:   LXI   H,ECCCHR      ;LOAD ADDRESS OF ECC CHAR.
4856  AE                7.0        XRA   M             ;EXCLUSIVE OR CHAR. AND ECC
4857  5F                4.0        MOV   E,A           ;SAVE XOR RESULT IN E
4858  E6  10            7.0        ANI   $10           ;IS BIT #4 OF RESULT SET
485A  7B                4.0        MOV   A,E           ;RESTORE XOR RESULT FROM B
485B  CA  60  48       10.0        JZ    ECC1          ;CONTINUE IF BIT #4 RESET
485E  EE  23            7.0        XRI   $23           ;ELSE-XOR WITH 23
4860  5F                4.0  ECC1: MOV   E,A           ;STORE THE ECC RESULT IN E
                                                        ;
4861  AF                4.0        XRA   A             ;CLEAR A
4862  4F                4.0        MOV   C,A           ;CLEAR THE TRANSLATE RESULT
4863  21  6E  48       10.0        LXI   H,ECCTBL      ;POINT TO ECC TABLE TO RE-POSITION
4866  CD  77  48       18.0        CALL  TRANS         ;TRANSLATE THE BITS
4869  79                4.0        MOV   A,C           ;GET THE TRANSLATED RESULT
486A  32  76  48       13.0        STA   ECCCHR        ;STORE RESULT
486D  C9               10.0        RET

                                ECCTBL: $08             ;BIT 0 = POSITION 3
                                        $20             ;BIT 1 = POSITION 5
                                        $02             ;BIT 2 = POSITION 1
                                        $40             ;BIT 3 = POSITION 6
                                        $80             ;BIT 4 = POSITION 7
                                        $01             ;BIT 5 = POSITION 0
                                        $10             ;BIT 6 = POSITION 4
                                        $04             ;BIT 7 = POSITION 2

4876  00                      ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

4877  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
4879  7B               4.0  TRANS1: MOV   A,E        ;GET CHAR TO BE TRANSLATED
487A  A0               4.0          ANA   B          ;SEE IF BIT POSITION IN 'B' IS SET
487B  CA  81  48       10.0         JZ    TRANS2     ;DO NEXT BIT POSITION IF NOT SET
487E  79               4.0          MOV   A,C        ;GET PREVIOUS RESULT OF 'OR'
487F  B6               7.0          ORA   M          ;'OR' IN NEW POSITION
4880  4F               4.0          MOV   C,A        ;SAVE RESULT

4881  78               4.0  TRANS2: MOV   A,B
4882  07               4.0          RLC              ;POSITION MASK TO NEXT BIT
4883  47               4.0          MOV   B,A
4884  D8               12.0         RC               ;EXIT WHEN ALL POSITIONS DONE
4885  23               6.0          INX   H          ;POINT TO NEXT TABLE ENTRY
4886  C3  79  48       10.0         JMP   TRANS1     ;PROCESS NEXT BIT
```

```
                                .SBTTL  SUBROUTINE TRK
                                SSUB
                                ;*****************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*GENERATE BAD TRACK TRANSLATION
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE GENRATES A TRACK NUMBER FOR THE BADT4X5 SUBROUTINE
                                ;*BASED ON THE VALUE OF INDAT.
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED      DATA BIT POSITION      TRACK WITH ECC
                                ;*          DATA            WITH ECC CORRECTION      CORRECTION
                                ;*      ------------        -------------------    -------------
                                ;*
                                ;*         XXXX0000                 0                    2
                                ;*         XXXX0001                 1                    8
                                ;*         XXXX0010                 2                    1
                                ;*         XXXX0011                 3                    9
                                ;*         XXXX0100                 4                    3
                                ;*         XXXX0101                 5                    5
                                ;*         XXXX0110                 6                    6
                                ;*         XXXX0111                 7                    7
                                ;*         XXXX1000                 P                    4
                                ;*         XXXX1001                 1                    8
                                ;*         XXXX1010                 2                    1
                                ;*         XXXX1011                 3                    9
                                ;*         XXXX1100                 4                    3
                                ;*         XXXX1101                 5                    5
                                ;*         XXXX1110                 6                    6
                                ;*         XXXX1111                 7                    7
                                ;*
                                ;*      X = DON'T CARE BITS
                                S
                                ;*****************************************************************************

                                TRK:    LDA     INDAT           ;GET INPUT DATA
                                        ANI     $0F             ;REMOVE 4 BITS
                                        CPI     $8              ;PARITY TRACK?
                                        JZ      TRKA            ;YES
                                        ANI     $7              ;NO-IGNORE BIT3
                                TRKA:   INR     A               ;INCREMENT FOR SUBROUTINE
                                        STA     BAD1            ;SAVE IT.
                                        DCR     A               ;PREPARE TO USE AS TABLE INDEX
                                        LXI     H,TIMTBL        ;TIE BUS TIME TABLE START
                                        MOV     E,A             ;LOW HALF OF TABLE INDEX
                                        MVI     D,0             ;ZERO HIGH HALF
                                        DAD     D               ;PRODUCE TABLE ENTRY ADDRESS
```

```
2677  48A1  7E           7.0          MOV     A,M          ;GET TIE BUS TIME FROM TABLE
2678  48A2  32 AF 48    13.0          STA     TIETIM       ;SAVE VALUE
2679  48A5  C9           10.0          RET
2680  48A6  02                  TIMTBL: .BYTE   2          ;TIE BUS TIME FOR BIT 0
2681  48A7  03                          .BYTE   3          ;TIE BUS TIME FOR BIT 1
2682  48A8  00                          .BYTE   0          ;TIE BUS TIME FOR BIT 2
2683  48A9  01                          .BYTE   1          ;TIE BUS TIME FOR BIT 3
2684  48AA  04                          .BYTE   4          ;TIE BUS TIME FOR BIT 4
2685  48AB  08                          .BYTE   8          ;TIE BUS TIME FOR BIT 5
2686  48AC  07                          .BYTE   7          ;TIE BUS TIME FOR BIT 6
2687  48AD  05                          .BYTE   5          ;TIE BUS TIME FOR BIT 7
2688  48AE  06                          .BYTE   6          ;TIE BUS TIME FOR BIT P
2689
2690
```

```
2692                                    .SBTTL  PROGRAM VARIABLES
2693
2694  48AF   00                         TIETIM: .BYTE   0               ;TIE BUS JAM TIME
2695  48B0   00                         INDAT:  .BYTE   0               ;DATA BYTE TO ECC
2696  48B1   00                         INDATP: .BYTE   0               ;DATA BYTE PARITY
2697  48B2   00                         UNITMP: .BYTE   0               ;UNIT MAP
2698  48B3   00                         SUNIT:  .BYTE   0               ;MASS BUS/TU SELECT BITS
2699  48B4   00   00                    AMTMSK: .WORD   0               ;AMTIE MASK WORD
2700  48B6   00                         DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2701  48B7   00                         DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2702  48B8   00                         DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2703  48B9   00                         DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2704         0000                               .END
```

```
A      -%0007      ADATA  = 0094      AMTIEP= 0001      AMTIE7= 0002
AMTMSK  48B4       ARAIDF- 0098      ASAVE   4F9B      ATTCD   4F97
AXNUM   4F91       B      =%0000      BAD     4773      BADST  = 0090
BADT4X  4764       BAD1    484B      BAD1C   47C0      BAD2    484C
BAD2C   47CB       BAD3    484D      BAD3C   47DD      BIT0   = 0001
BIT1   = 0002      BIT15  = 8000      BIT2   = 0004      BIT3   = 0008
BIT4   = 0010      BIT5   = 0020      BIT6   = 0040      BIT7   = 0080
BIT8   = 0100      BIT9   = 0200      BRKPBC= 4F0A      BRKRAM= 4F10
BRKSTR= 4E60       BRKXCT= 4F00      BSAVE   4F9C      BYTCNT= 00D4
BYTEH   4F24       BYTEL   4F23      B4X5    4784      C      =%0001
CASCT   4F21       CASCTL= 00A0      CASSTA= 00A0      CATTH  = 0089
CATTL  = 0088      CBUSST= 00A1      CBYTH  = 008B      CBYTL  = 008A
CDG1H  = 0087      CDG1L  = 0086      CDG2H  = 0093      CDG2L  = 0092
CDG3H  = 0095      CDG3L  = 0094      CDVTH  = 008D      CDVTL  = 008C
CHPTIE= 0028       CHOTIE= 0020      CH1TIE= 0021      CH2TIE= 0022
CH3TIE= 0023       CH4TIE= 0024      CH5TIE= 0025      CH6TIE= 0026
CH7TIE= 0027       CKLOP  = 2815      CLEAR   4739      CLECC   484E
CLKCTL= 00F0       CLOCK   4F26      CLRLP   473D      CMC0H  = 0099
CMC0L  = 0098      CMC1H  = 009B      CMC1L  = 009A      CMC2H  = 009D
CMC2L  = C09C      CMC3H  = 009F      CMC3L  = 009E      CMINH  = 0097
CMINL  = 0096      CNTCTL= 00D7      CRCWRD= 0018      CSAVE   4F9D
CSRLH  = 0091      CSRLL  = 0090      CTCH   = 0085      CTCL   = 0084
CTSTH  = 008F      CTSTL  = 008E      CXCTH  = 0081      CXCTL  = 0080
CXINH  = 0083      CXINL  = 0082      C.     = 0001      C.AVAI= 0080
C.DP   = 0008      C.DSE  = 0010      C.DTU  = 0003      C.DVA  = 0008
C.FAIL= 00FC       C.FMT  = 0070      C.FNCT= 003E      C.GO   = 0001
C.INTC= 00FE       C.MAIN= 0020      C.NSA  = 0080      C.RCT  = 00FC
C.SER  = 0080      C.SHR  = 0040      C.SKPC= 000F      C.TAPE= 0040
C.WCS  = 0002      C4X5    478D      D      =%0002      DATAA   48B6
DATAAP  48B7       DATACT= 00D0      DATAE   48B8      DATAEP  48B9
DBUS    4F28       DBUSCT= 00C0      DBUSST= 00C0      DDRA   = 00D8
DDRAIN= 0010       DDRB   = 00D9      DDRBIN= 0002      DDRC   = 00DA
DDRCIN= 0001       DDRCO  = 0088      DDRCTL= 00DB      DIAFLG  4F22
DIAGPG= 4300       DIAGRM= 4F90      DIARD  = 0008      DONE1  = 0045
DONINT= 0010       DSAVE   4F9E      DSE    = 0006      DUMMU   460C  G
DUMMV   4652  G    DUMMW   444C  G   DUMMX   4419  G   DUMMY   431E  G
D.ATH0= 0001       D.ATH1= 0002      D.EOTD= 0010      D.LAGC= 0020
D.NOTW= 0040       D.NTHR= 0004      D.TACH= 0008      D.WR4  = 0080
D4X5    477C       E      =%0003      ECC     4853      ECCBAD= 0042
ECCCHR  4876       ECCCOR= 0019      ECCOK  = 0041      ECCSTA= 001A
ECCTBL  486E       ECCTST= 000E      ECC1    4860      EDATA  = 0095
EOTCLR= 0003       ERFLG   4F93      ERLP   = 2809      ERLPA  = 280F
ERLPB  = 2812      ERLPE  = 280C      ERNUM   4F90      ERRCNT= 00D6
ESAVE   4F9F       EXIT    4736      E.ACRC= 0010      E.AMT  = 0020
E.CDP  = 0080      E.CRC  = 0080      E.PNTR= 0008      E.RPE  = 0040
E.STEC- 0001       E.TTEC= 0002      E.UNC  = 0004      E4X5    47A6
FIFORD= 006A       FORMAT  4F25      FOUND   4344      FWDTST= 0061
F4X5    47F2       GCRID  = 0089      GCRSET= 0002      GOODTM= 0092
GP4X5   4849       H      =%0004      HLSAVE  4FA0      IE     = 0008
INDAT   48B0       INDATP  48B1      INTSTA= 00E0      INV4X5  47D6
ITERA   4F9A       I.PWR  = 0020      I.RMPE= 0040      I5.5   = 0010
I6.5   = 0020      I7.5   = 0040      KCALL  = 005F      KCLR   = 007B
KDEP   = 003F      KENAB  = 0078      KEXAM  = 003E      KEYBRD= 00C8
KEY1   - 0078      KEY10  = 006D      KEY11  = 006E      KEY12  = 006F
```

```
KEY13 = 005C    KEY14 = 005D    KEY15 = 005E    KEY16 = 005F
KEY17 = 003C    KEY18 = 003D    KEY19 = 003E    KEY2  = 0079
KEY20 = 003F    KEY3  = 007A    KEY4  = 007B    KEY5  = 0074
KEY6  = 0075    KEY7  = 0076    KEY8  = 0077    KEY9  = 006C
KINTA = 006F    KLDAD = 003D    KN0   = 003C    KN1   = 005C
KN2   = 005D    KN3   = 005E    KN4   = 006C    KN5   = 006D
KN6   = 006E    KN7   = 0074    KN8   = 0075    KN9   = 0076
KU2   = 0079    KU3   = 007A    KU8   = 0077    L     =%0005
LBLANK= 000F    LCE   = 000B    LCH   = 000C    LCL   = 000D
LCP   = 000E    LC0   = 0000    LC1   = 0001    LC2   = 0002
LC3   = 0003    LC4   = 0004    LC5   = 0005    LC6   = 0006
LC7   = 0007    LC8   = 0008    LC9   = 0009    LDLEDA= 00CA
LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD    LDLEDE= 00CE
LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C    LKKEY = 0049
LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052    LKLWPP= 004F
LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94    LPNUM   4F92
M     =%0006    MBSEL = 00E0    MB.A  = 0008    MB.B  = 0004
MEMTOP= 4FFF    MINUS = 000A    MM    = 8000    MSE   = 0008
MSGN  = 000F    MTACLR= 0000    MT.ARA= 0020    MT.CPE= 0080
MT.DSE- 0001    MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004
MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004
MT.PS0- 0001    MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010
MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020    M.CONT= 0080
M.DEM = 0020    M.EBL = 0010    M.EXC = 0008    M.FAIL= 0008
M.ILR - 0010    M.INIT= 0010    M.OCC = 0020    M.ONLI= 0001
M.PF  = 0040    M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008
M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040    M.UNIT= 0007
M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080    M5.5  = 0001
M6.5  = 00C2    M7.5  = 0004    NOTCAP= 0088    OKAY  = 00FF
OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040    PADCNT= 00D5
PADCRC= 0080    PDIAG = 0048    PEID  = 008A    PENAB = 004C
PESET = 0001    PL    = 00B1    POINTB  4827    POINTE  480F
PO4X5   4793    PRDD  = 004C    PRENF = 009C    PS    = 00B2
PSTAT = 0048    PSW   =%0009    P.AMTP= 0001    P.BCTC= 0040
P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040    P.LWR = 0020
P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002    P.RPOE= 0020
P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080
P.STAT= 0002    P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010
P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001    P.WPEN= 0010
P.WPOE- 0008    P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004
P.5VOK= 0002    QUE   = 281B    QUEM  - 281E    RAMT  = 0010
RARA    0006    RARAI = 0004    RCHBD0= 0048    RCHBD1= 0047
RCHOK = 0046    RCHTST= 000C    RCLRT = 000D    RCMD  = 0008
RCMLP = 0003    RCONT = 0080    RDATA = 0017    RDCLK = 0010
RDON  = 0011    READG = 0007    REND  = 0014    REQST = 2806
RESCHR= 00D1    REVTST= 0064    REWIND= 0004    RFIFOL= 0008
RGCLK = 0002    RGCRI = 0003    RIBG  = 0001    RILL  = 0012
RINST = 000C    RMCTST= 0008    RMK2  = 0013    RNOP  = 0000
RPATH = 0001    RPBAD = 0044    RPCHI = 0001    RPCLK = 0003
RPCTL - 0009    RPEI  - 0002    RPFAIL= 0000    RPF1  = 009D
RPF2  = 009E    RPOK  = 0043    RPOSTN= 0016    RPSTA = 0015
RRCMT = 000A    RSTAT = 0002    RTIEB - 000A    RTIER = 0030
RTM   = 0005    RUNKI = 0009    RUPTST= 005E    RWDUNL= 0005
R.AMT - 0001    R.BOP = 0008    R.DATA- 0040    R.DON = 0002
```

```
R.DRDY= 0010        R.END = 0010        R.ILL = 0004        R.JVOK= 0004
R.MK2 = 0008        R.PLOD= 0008        R.PLOO= 0010        R.PLO1= 0020
R.POST= 0020        R.STNM= 0002        R.STOP= 0004        R.STPC= 0001
R.TBJN= 0080        R.TSTD= 0040        R.VOK = 0080        R00H  = 0081
R00L  = 0080        R01H  = 0083        R01L  = 0082        R02H  = 0085
R02L  = 0084        R03H  = 0087        R03L  = 0086        R04H  = 0089
R04L  = 0088        R05H  = 008B        R05L  = 008A        R06H  = 008D
R06L  = 008C        R07H  = 008F        R07L  = 008E        R10H  = 0091
R10L  = 0090        R11H  = 0093        R11L  = 0092        R12H  = 0095
R12L  = 0094        R13H  = 0097        R13L  = 0096        R14H  = 0099
R14L  = 0098        R15H  = 009B        R15L  = 009A        R16H  = 009D
R16L  = 009C        R17H  = 009F        R17L  = 009E        R7.5  = 0010
SELCLR= 0000        SETATA= 00A1        SID   = 0080        SOD   = 0080
SOE   = 0040        SP    =%0008        SSCLK = 0040        SSTEP = 0005
STACK = 4FFF        STATRM= 4F20        STPCT   4F20        STRSP = 5000
SUNIT   48B3        TAB4X5  482B        TADR00= 0080        TADR01= 0081
TADR02= 0082        TADR03= 0083        TADR04= 0084        TADR05= 0085
TADR06= 0086        TADR07= 0087        TADR10= 0088        TADR11= 0089
TADR12= 008A        TADR13= 008B        TAMT  = 0044        TASEL = 0080
TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004
TC.REV= 0020        TC.WRT= 0010        TEMP    4F99        TEST1   4300
TEST2   4533        TIETIM  48AF        TIMTBL  48A6        TMF   = 0099
TMRDY = 0040        TRANS   4877        TRANS1  4879        TRANS2  4881
TRK     4889        TRKA    4895        TRKENA= 00D2        TRNIN   483B
TRNOUT  483F        TRNTMP  484A        TSET  = 2803        TSTEND= 2818
TSTS  = 0040        TST01A  4422        TST01C  4379        TST01F  4391
TST01G  438A        TST01H  43E8        TST01I  439D        TST01K  4455
TST01L  446B        TST01M  4483  G     TST01N  44CD        TST01Q  44AE
TST01R  44A5  G     TST01S  44C2        TST01T  44BC  G     TST01U  44EA  G
TST01V  450C  G     TST01X  4351        TST01Y  4457        TST02A  4615
TST02B  4619        TST02C  456F        TST02F  4587        TST02G  45AD
TST02H  45DB        TST02I  4593        TST02K  465B        TST02L  4671
TST02M  4689  G     TST02N  46D3        TST02Q  46B4        TST02R  46AB  G
TST02S  46C8        TST02T  46C2  G     TST02U  46F0  G     TST02V  4712  G
TST02X  4542        TST02Y  465D        TUSEL0= 00D1        TUSEL1= 00D2
TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002        T.BOT = 0004
T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020
T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008        T.PS1J= 0010
T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010        T.SCLK= 0002
T4X5    4768        UIBG  = 00A1        UNITMP  48B2        UNITSL  4327
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN- 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT - 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X     -%000A
X.DONN- 0080        X.ENAB  0040        X.PEPE= 0002        X.ROME= 0001
X.WCLK  0001        Y     =%000B        .     - 48BA
```

ERRORS DETECTED: 0

```
*ECC2.A78/PTP,ECC2-NLIST,PARAM,MACRO,LIST,ECC2
RUN-TIME: 4 6 0 SECONDS
CORE USED:  10K
```

```
                                .TITLE  ECC3 - ECC CONTROLLER PART #3
                                .SBTTL  TEST 1 - ECC BAD DATA ONE TRACK INCORRECT POINTER
                                ;ID     ECC3-ERROR CORRECTION CONTROLLER PART #3
 4300                           ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ECC BAD DATA ONE TRACK INCORRECT POINTER
 4300                           SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM SINGLE
                                ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN A SINGLE TRACK, AND A
                                ;*INCORRECT POINTER ALSO PROVIDED.  THE PROGRAM CHECKS THAT THE DATA IS
                                ;*CORRECTED, AND THAT THE SINGLE TRACK ERROR CORRECT STATUS BIT AND THE
                                ;*POINTER ERROR STATUS BIT ARE SET.
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND THE DATA BIT POSITION OF THE INCORRECT POINTER
                                ;*CAN BE DETERMINED FROM THE FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITION       BIT POSITION OF
                                ;*          DATA                WITH ECC CORRECTION       INCORRECT POINTER
                                ;*      --------------          -------------------     --------------
                                ;*
                                ;*          XXXX0000                    0                       1
                                ;*          XXXX0001                    1                       2
                                ;*          XXXX0010                    2                       3
                                ;*          XXXX0011                    3                       4
                                ;*          XXXX0100                    4                       5
                                ;*          XXXX0101                    5                       6
                                ;*          XXXX0110                    6                       7
                                ;*          XXXX0111                    7                       P
                                ;*          XXXX1000                    P                       0
                                ;*          XXXX1001                    1                       2
                                ;*          XXXX1010                    2                       3
                                ;*          XXXX1011                    3                       4
                                ;*          XXXX1100                    4                       5
                                ;*          XXXX1101                    5                       6
                                ;*          XXXX1110                    6                       7
                                ;*          XXXX1111                    7                       P
                                ;*
                                ;*      X = DON'T CARE BITS
                                ;*
                                ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                                ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED
                                ;*          DATA                        POINTER GENERATED
                                ;*      --------------                  -----------------
                                ;*
                                ;*          XX00XXXX                    HISTORY
                                ;*          XX01XXXX                    PHASE TIE
```

```
                                ;*          XX10XXXX                    AMTIE
                                ;*          XX11XXXX                    ILLEGAL 5X4
                                ;*
                                ;*       X = DON'T CARE BITS
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*     CLEAR THE INPUT DATA
                                ;*     CLEAR BAD TRANSLATION POINTERS
                                ;*   BGNDO
                                ;*   : ENABLE THE READ PATH CLOCK
                                ;*   : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*   : CLOCK THE FIFO'S
                                ;*   : ISSUE CLEAR ALL COMMAND
                                ;*   : WAIT
                                ;*   : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                                ;*   : CALL SUBROUTINE CLECC
                                ;*   : SET THE LOOP COUNT TO 7
                                ;*   : BGNDO
                                ;*   : :   GET THE INPUT DATA BYTE
                                ;*   : :   CALL SUBROUTINE ECC
                                ;*   : :   DECREMENT THE LOOP COUNT
                                ;*   : :   DO UNTIL THE LOOP COUNT=0
                                ;*   : ENDDO
                                ;*   : SET THE LOOP COUNT TO 3
                                ;*   : BGNDO
                                ;*   : :   FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
                                ;*   : :   CALL SUBROUTINE TRK
                                ;*   : :   CALL SUBROUTINE BADT4X5
                                ;*   : :   INIT THE LOOP COUNT TO 5
                                ;*   : :   BGNDO
                                ;*   : :   :   MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                                ;*   : :   :   INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*   : :   :   MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                                ;*   : :   :   CLOCK THE DATA INTO THE FIFO
                                ;*   : :   :   DECREMENT THE LOOP COUNT
                                ;*   : :   :   INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*   : :   :   DO UNTIL LOOP COUNT = 0
                                ;*   : :   ENDDO
                                ;*   : :   FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
                                ;*   : :   FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
                                ;*   : :   CALL SUBROUTINE TRK
                                ;*   : :   CALL SUBROUTINE BADT4X5
                                ;*   : :   INIT THE LOOP COUNT TO 5
                                ;*   : :   BGNDO
                                ;*   : :   :   MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
                                ;*   : :   :   MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
```

```
                                ;*  : : :  CLOCK THE DATA INTO THE FIFO
                                ;*  : : :  DECREMENT THE LOOP COUNT
                                ;*  : : :  DO UNTIL THE LOOP COUNT = 0
                                ;*  : : ENDDO
                                ;*  : : DECREMENT THE LOOP COUNT
                                ;*  : : DO UNTIL THE LOOP COUNT = 0
                                ;*  : ENDDO
                                ;*  : ISSUE DIAGNOSTIC READ COMMAND
                                ;*  : SET UP WATCHDOG TIMER COUNT
                                ;*  : BGNDO
                                ;*  : :    SINGLE STEP THE READ PATH
                                ;*  : :    DECREMENT THE WATCHDOG TIMER
                                ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
                                ;*  : ENDDO
                                ;*  : IF WATCHDOG TIMER=0
                                ;*  : :    THEN-ERROR
                                ;*  : :    ELSE-CONTINUE
                                ;*  : ENDIF
                                ;*  : GET THE TIE BUS TABLE ADDRESS
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : INIT THE LOOP COUNT TO 0
                                ;*  : BGNDO
                                ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT
                                ;*  : :    : THEN-CALL SUBROUTINE POINTER
                                ;*  : :    : ELSE-CLEAR TIE BUS POINTER
                                ;*  : :    ENDIF
                                ;*  : :    LOAD THE TIE BUS REGISTER
                                ;*  : :    SINGLE STEP THE READ PATH
                                ;*  : :    DECREMENT THE LOOP COUNT
                                ;*  : :    DO UNTIL LOOP COUNT = 9
                                ;*  : ENDDO
                                ;*  : SET UP WATCHDOG TIMER COUNT
                                ;*  : BGNDO
                                ;*  : :  SINGLE STEP THE READ PATH
                                ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
                                ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
                                ;*  : ENDDO
                                ;*  : IF WATCHDOG TIMER=0
                                ;*  : :  THEN-ERROR-EXIT TEST
                                ;*  : :  ELSE-CONTINUE
                                ;*  : ENDIF
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : INIT THE  LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  SINGLE STEP THE READ PATH
                                ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
                                ;*  : :  IF NOT EQUAL
                                ;*  : :  : THEN-ECC ERROR
                                ;*  : :  : ELSE-CONTINUE
                                ;*  : :  ENDIF
                                ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
                                ;*  : :  IF NOT EQUAL
                                ;*  : :  : THEN-ECC ERROR
                                ;*  : :  : ELSE-CONTINUE
```

```
                                ;*   : :  ENDIF
                                ;*   : :  DECREMENT THE LOOP COUNT
                                ;*   : :  DO UNTIL THE LOOP COUNT = 0
                                ;*   : ENDDO
                                ;*   : SINGLE STEP THE READ PATH
                                ;*   : IF EXPECTED NOT EQUAL
                                ;*   : :  THEN-ECC ERROR
                                ;*   : :  ELSE-CONTINUE
                                ;*   : ENDIF
                                ;*   : INPUT THE ECC STATUS
                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*   : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
                                ;*   : :  THEN-CONTINUE
                                ;*   : :  ELSE-ERROR
                                ;*   : ENDIF
                                ;*   : INCREMENT THE INPUT DATA
                                ;*   : DO UNTIL INPUT DATA = 0
                                ;*  ENDDO
                                ;*ENDTST
4300                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*ECC3 MICRO TEST 01
                                ;*ECC3 MICRO ERROR 01
                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                ;*M8951, M8950'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*ECC3 MICRO TEST 01
                                ;*ECC3 MICRO ERROR 02
                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                ;*'BYTE/SCLK COUNT NUMBER = LLL''
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC3 MICRO TEST 01
                                ;*ECC3 MICRO ERROR 03
                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                                ;*SENT TO THE ECC
                                ;*'BYTE/SCLK COUNT NUMBER = LLL''
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC3 MICRO TEST 01
                                ;*ECC3 MICRO ERROR 04
                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                ;*M8951, M8950'S
                                ;*CORRECTED DATA FROM THE ECC-INCORRECT
                                ;*'BYTE/SCLK COUNT NUMBER = LLL''
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 05
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL''
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 06
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL''
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 07
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL''
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 10
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL''
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
(1)                                     ; ***************************************************************************
1566
1567 4300                               TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1) 4300  3E  01              7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28         18.0                 CALL    TSET            ;SETUP THE TEST
1568                                    ;%ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
1569                                    ;&M8951, M8950'S
1570
1571 4305                                       REQ     @7,0,0,0,0
 (1) 4305  CD  06  28         18.0                 CALL    REQST
 (1) 4308  00                                      .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4309  00  00                                  .WORD   0               ;SYSTEM ''0'' COUNT
 (1) 430B  00  00                                  .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D  00                                      .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1) 430E  07                                      .BYTE   @7              ;REQUEST CODE
1572 430F                                       RIN     R12L
 (1) 430F  DB  94              10.0                IN      R12L            ;READ R12L INTO AC
 (1) 4311  7F                   4.0                MOV     A,A             ;RETRY LINK
1573 4312  32  BC  46          13.0         STA     UNITMP
1574 4315  A7                   4.0         ANA     A
1575 4316  C2  27  43          10.0         JNZ     UNITSL
```

```
1576  4319                                        ERR     EXIT,DUMMY
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                           .BYTE
 (1)  431E  CD  15  28      18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  40  45      10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1577                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1578                                        ;>FATAL ERROR - TEST ABORTED
1579  4324  C3  40  45      10.0                JMP     EXIT
1580
1581  4327  06  00            7.0  UNITSL: MVI     B,0
1582  4329  3A  BC  46       13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1583  432C  E6  01            7.0           ANI     @01             ;PORT 0?
1584  432E  C2  44  43       10.0           JNZ     FOUND           ;YES-GO USE PORT #0
1585  4331  04                4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
1586  4332  3A  BC  46       13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1587  4335  E6  02            7.0           ANI     @02             ;PORT 1?
1588  4337  C2  44  43       10.0           JNZ     FOUND           ;YES-GO USE PORT #1
1589  433A  04                4.0           INR     B               ;NO-UPDATE POINTER TO PORT #2
1590  433B  3A  BC  46       13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1591  433E  E6  04            7.0           ANI     @04             ;PORT 2?
1592  4340  C2  44  43       10.0           JNZ     FOUND           ;YES-GO USE PORT #2
1593  4343  04                4.0           INR     B               ;NO-ASSUME PORT #3
1594  4344  CD  43  45       18.0  FOUND:   CALL    CLEAR           ;CLEAR ALL TU PORTS
1595  4347  DB  E0           10.0           IN      INTSTA
1596  4349  E6  80            7.0           ANI     BIT7
1597  434B  B0                4.0           ORA     B
1598  434C  D3  E0           10.0           OUT     MBSEL
1599
1600  434E  3E  10            7.0  TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1601  4350  D3  F0           10.0           OUT     CLKCTL
1602  4352  3E  10            7.0           MVI     A,W.GCR
1603  4354  D3  D3           10.0           OUT     WMCCTL
1604  4356  3E  04            7.0           MVI     A,P.RPEN
1605  4358  D3  4C           10.0           OUT     PENAB
1606
1607  435A  3E  60            7.0           MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1608  435C  D3  48           10.0           OUT     PDIAG           ;LCS, LWR AND DPEN
1609  435E  AF                4.0           XRA     A
1610  435F  D3  44           10.0           OUT     TAMT
1611  4361  D3  D2           10.0           OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1612  4363  D3  0A           10.0           OUT     RTIEB           ;CLEAR THE TIE BUS REGISTER
1613  4365  D3  D2           10.0           OUT     TRKENA
1614  4367  32  55  46       13.0           STA     BAD1            ;CLEAR THE INVALID
1615  436A  32  56  46       13.0           STA     BAD2            ;TRACK TRANSLATIONS
1616  436D  32  57  46       13.0           STA     BAD3            ;POINTERS
1617  4370  32  BA  46       13.0           STA     INDAT           ;CLEAR THE INPUT DATA BYTE
1618  4373                                   ROUT    R05H
 (1)  4373  D3  8B           10.0                   OUT     R05H            ;WRITE AC INTO R05H
.(1)  4375  7F                4.0                   MOV     A,A             ;RETRY LINK
1619
1620  4376  3E  A8            7.0  TST01C: MVI     A,R.PL01!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
```

```
1621                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1622  4378  D3  09          10.0          OUT     RPCTL
1623  437A  D3  08          10.0          OUT     RFIFOL                 ;CLOCK THE FIFO'S
1624  437C  3E  0D           7.0          MVI     A,RCLRT                ;ISSUE CLEAR ALL COMMAND
1625  437E  D3  0B          10.0          OUT     RCMD
1626
1627  4380  00               4.0          NOP                            ;WAIT
1628  4381  00               4.0          NOP
1629  4382  00               4.0          NOP
1630  4383  00               4.0          NOP
1631  4384  00               4.0          NOP
1632
1633  4385  3E  A9           7.0          MVI     A,R.PLO1.R.STPC!R.PLOD.R.TBJN  ;STOP THE READ PATH
1634                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1635  4387  D3  09          10.0          OUT     RPCTL
1636  4389  CD  58  46      18.0          CALL    CLECC                  ;CLEAR THE ECC CHARACTER
1637  438C  16  07           7.0          MVI     D,@7                   ;SET UP THE LOOP COUNT
1638  438E  3A  BA  46       13.0  TST01F: LDA     INDAT                  ;GET THE INPUT DATA
1639  4391  CD  5D  46       18.0          CALL    ECC                    ;CALCULATE ECC FOR
1640  4394  15               4.0          DCR     D                      ;THE 7 INPUT BYTES
1641  4395  C2  8E  43      10.0          JNZ     TST01F
1642
1643  4398  0E  03           7.0          MVI     C,@3                   ;INIT THE LOOP COUNT
1644  439A  3A  BA  46       13.0  TST01I: LDA     INDAT                  ;GET THE INPUT DATA
1645  439D                                ROUT    R05L
 (1)  439D  D3  8A          10.0                  OUT     R05L                   ;WRITE AC INTO R05L
 (1)  439F  7F               4.0                  MOV     A,A                    ;RETRY LINK
1646  43A0  32  45  46       13.0          STA     TRNIN                  ;FILL THE TRANSLATOR
1647  43A3  32  46  46       13.0          STA     TRNIN+1                ;SUBROUTINE INPUT
1648  43A6  32  47  46       13.0          STA     TRNIN+2                ;BUFFER
1649  43A9  32  48  46       13.0          STA     TRNIN+3                ;
1650  43AC  CD  93  46       18.0          CALL    TRK
1651  43AF  CD  6E  45       18.0          CALL    BADT4X5                ;TRANSLATE THE SUBGROUP
1652
1653  43B2  21  49  46       10.0          LXI     H,TRNOUT               ;GET POINTER TO TRANSLATE DATA TABLE
1654  43B5  06  05           7.0          MVI     B,@5                   ;SET UP LOOP COUNT
1655  43B7  7E               7.0  TST01G: MOV     A,M                    ;GET A DATA BYTE
1656  43B8  D3  40          10.0          OUT     TCMD                   ;STORE DATA IN COMMAND ADDRESS
1657  43BA  23               6.0          INX     H                      ;POINT TO DATA PARITY
1658  43BB  7E               7.0          MOV     A,M                    ;GET THE DATA PARITY
1659
1660  43BC  07               4.0          RLC                            ;POSITION FOR OUTPUT
1661  43BD  F6  60           7.0          ORI     P.LWR!P.LCS            ;OR IN CONTROL BITS
1662  43BF  D3  48          10.0          OUT     PDIAG                  ;OUTPUT THE DATA PARITY
1663
1664  43C1  D3  08          10.0          OUT     RFIFOL                 ;CLOCK DATA INTO THE FIFO'S
1665
1666  43C3  23               6.0          INX     H                      ;UPDATE THE TABLE POINTER
1667  43C4  05               4.0          DCR     B                      ;DECREMENT LOOP COUNT
1668  43C5  C2  B7  43      10.0          JNZ     TST01G                 ;DO UNTIL LOOP COUNT = 0
1669
1670  43C8  3A  BA  46       13.0          LDA     INDAT                  ;GET THE INPUT DATA
1671  43CB  32  45  46       13.0          STA     TRNIN                  ;FILL THE FIRST THREE
1672  43CE  32  46  46       13.0          STA     TRNIN+1                ;BYTES OF THE TRANSLATOR
```

```
1673  43D1  32  47  46      13.0          STA     TRNIN+2         ;SUBROUTINE WITH THE INPUT DATA
1674  43D4  3A  80  46      13.0          LDA     ECCCHR          ;STORE THE CALCULATED
1675  43D7  32  48  46      13.0          STA     TRNIN+3         ;ECC CHARACTER AS THE LAST CHARACTER
1676  43DA  CD  93  46      18.0          CALL    TRK
1677
1678  43DD  CD  6E  45      18.0          CALL    BADT4X5         ;TRANSLATE THE SECOND SUBGROUP
1679
1680  43E0  21  49  46      10.0          LXI     H,TRNOUT        ;GET THE POINTER TO INPUT DATA TABLE
1681  43E3  06  05           7.0          MVI     B,@5            ;SET UP THE LOOP COUNT
1682  43E5  7E               7.0  TST01H: MOV     A,M             ;GET A DATA BYTE
1683  43E6  D3  40          10.0          OUT     TCMD            ;STORE DATA IN CMD ADDRESS
1684  43E8  23               6.0          INX     H               ;POINT TO DATA PARITY BIT
1685  43E9  7E               7.0          MOV     A,M             ;GET THE DATA PARITY
1686  43EA  07               4.0          RLC                     ;POSITION FOR OUTPUT
1687  43EB  F6  60           7.0          ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
1688  43ED  D3  48          10.0          OUT     PDIAG           ;OUTPUT THE DATA PARITY
1689  43EF  D3  08          10.0          OUT     RFIFOL          ;CLOCK DATA INTO THE FIFOS
1690  43F1  23               6.0          INX     H               ;UPDATE THE TABLE POINTER
1691  43F2  05               4.0          DCR     B               ;DECREMENT THE LOOP COUNT
1692  43F3  C2  E5  43      10.0          JNZ     TST01H          ;DO UNTIL LOOP COUNT=0
1693
1694  43F6  0D               4.0          DCR     C               ;PUT THE GROUP IN
1695  43F7  C2  9A  43      10.0          JNZ     TST01I          ;THE FIFO TWICE
1696
1697  43FA  3E  0B           7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
1698  43FC  D3  0B          10.0          OUT     RCMD            ;COMMAND
1699  43FE  11  01  00      10.0          LXI     D,1             ;SET WATCH DOG INCREMENT
1700  4401  21  A8  FD      10.0          LXI     H,-600          ;SET WATCH DOG COUNT TO 600
1701  4404  D3  0C          10.0  2$:     OUT     RINST           ;STEP THE READ PATH
1702  4406  DB  01          10.0          IN      RPCHI           ;DATA READY SET?
1703  4408  E6  10           7.0          ANI     R.DRDY
1704  440A  C2  1F  44      10.0          JNZ     TST01A          ;YES-GO PROCESS
1705  440D  19               10.0         DAD     D               ;WATCH DOG TIMEOUT?
1706  440E  D2  04  44       10.0         JNC     2$              ;NO-CONTINUE
1707  4411                                ERR     TST01C,DUMMU,1
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4411  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4414  02                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4415  01                           .BYTE   1
 (1)  4416  CD  15  28      18.0  DUMMU:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4419  DA  76  43      10.0          JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1708                                ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1709                                ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1710                                ;<FATAL ERROR - MICRO TEST ABORTED
1711  441C  C3  40  45      10.0          JMP     EXIT
1712
1713  441F  D3  0C          10.0  TST01A: OUT     RINST
1714  4421  06  00           7.0          MVI     B,$0            ;INIT THE BIT TIME COUNT (LOOP COUNT)
1715  4423  00               4.0  TST01B: NOP
1716  4424  3A  B9  46      13.0          LDA     TIETIM          ;GET THE TIME FOR TIE BUS JAMMING (INCORRECT POINTER)
1717  4427  B8               4.0          CMP     B               ;IS IT TIME TO JAM THE TIE BUS?
1718  4428  C2  31  44      10.0          JNZ     4$              ;NO - CONTINUE
1719  442B  CD  19  46      18.0          CALL    POINTER         ;YES - GO GET A REAL POINTER
```

```
1720  442E  C3  34  44   10.0          JMP   5$                ;
1721  4431  AF               4.0   4$:  XRA   A                 ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
1722  4432  D3  0A          10.0         OUT   RTIEB             ;LOAD THE POINTER
1723  4434  D3  0C          10.0   5$:   OUT   RINST             ;STEP CLOCK
1724  4436  04               4.0         INR   B                 ;DECREMENT THE LOOP COUNT
1725  4437  78               4.0         MOV   A,B
1726  4438  FE  09           7.0         CPI   $9                ;DONE
1727  443A  C2  23  44        10.0         JNZ   TST01B           ;DO UNTIL LOOP COUNT = ZERO
1728  443D  AF               4.0         XRA   A
1729  443E  D3  0A          10.0         OUT   RTIEB             ;CLEAR THE TIE BUS REGISTER
1730  4440  D3  0C          10.0         OUT   RINST             ;CLOCK THE READ PATH
1731  4442  D3  0C          10.0         OUT   RINST             ;CLOCK THE READ PATH
1732  4444  11  01  00       10.0         LXI   D,1               ;SET WATCHDOG TIMER INCREMENT
1733  4447  21  A8  FD       10.0         LXI   H,-600            ;SET WATCHDOG COUNT TO 600
1734  444A  D3  0C          10.0   3$:   OUT   RINST             ;SINGLE STEP THE READ PATH
1735  444C  DB  01          10.0         IN    RPCHI             ;DATA READY SET?
1736  444E  E6  10           7.0         ANI   R.DRDY
1737  4450  C2  65  44        10.0         JNZ   TST01K           ;YES-GO PROCESS
1738  4453  19               10.0         DAD   D                 ;NO-WATCHDOG TIMEOUT?
1739  4454  D2  4A  44        10.0         JNC   3$                ;NO-CONTINUE
1740  4457                               ERR   TST01C,DUMMV,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4457  CD  09  28        18.0                  CALL  ERLP              ;PROCESS ERROR - DO 2.3
 (1)       0003                         MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  445A  03                         .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  445B  01                         .BYTE 1
 (1)  445C  CD  15  28        18.0   DUMMV:: CALL  CKLOP             ;CHECK LOOP FUNCTION - DO 2.3
 (1)  445F  DA  76  43        10.0           JC    TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1741                          ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1742                          ;>SENT TO THE ECC
1743                          ;<FATAL ERROR - MICRO TEST ABORTED
1744  4462  C3  40  45        10.0         JMP   EXIT
1745
1746  4465  16  01             7.0   TST01K: MVI   D,@1            ;INIT THE LOOP COUNT
1747  4467  AF               4.0   TST01Y: XRA   A
1748  4468                               ROUT  R05H
 (1)  4468  D3  8B          10.0                 OUT   R05H              ;WRITE AC INTO R05H
 (1)  446A  7F               4.0                 MOV   A,A               ;RETRY LINK
1749  446B  3A  BA  46        13.0         LDA   INDAT            ;GET THE INPUT DATA
1750  446E                               ROUT  EDATA
 (1)  446E  D3  95          10.0                 OUT   EDATA             ;WRITE AC INTO EDATA
 (1)  4470  7F               4.0                 MOV   A,A               ;RETRY LINK
1751  4471  A7               4.0         ANA   A                 ;SET CONDITION BITS
1752  4472  3E  00             7.0         MVI   A,0               ;CLEAR THE ACC.
1753  4474  E2  78  44        10.0         JPO   1$                ;ODD # OF ONES - CONTINUE
1754  4477  3C               4.0         INR   A                 ;EVEN # OF ONES - SET PARITY
1755  4478  32  BB  46        13.0   1$:   STA   INDATP           ;STORE THE PARITY BIT
1756  447B  D3  0C          10.0   TST01L: OUT   RINST            ;SINGLE STEP THE READ LOGIC
1757  447D  3A  BA  46        13.0         LDA   INDAT
1758  4480  47               4.0         MOV   B,A
1759  4481  DB  19          10.0         IN    ECCCOR
1760  4483  B8               4.0         CMP   B
1761  4484  CA  93  44        10.0         JZ    TST01M
1762  4487                               ROUT  ADATA
```

```
 (1)  4487  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4489  7F               4.0                MOV     A,A             ;RETRY LINK
1763  448A  78               4.0        MOV     A,B             ;GET THE BYTE COUNT
1764  448B                                 ROUT    R05L            ;STORE IN CAS
 (1)  448B  D3  8A          10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  448D  7F               4.0                MOV     A,A             ;RETRY LINK
1765  448E                                 ERRB    TST01C,TST01M,@1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  448E  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4491  04                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4492  01                                      .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  4493  CD  15  28      18.0                TST01M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4496  DA  76  43      10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1766                                 ;>CORRECTED DATA FROM THE ECC-INCORRECT
1767  4499  DB  15          10.0        IN      RPSTA           ;GET THE ECC PARITY BIT
1768  449B  E6  80           7.0        ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
1769  449D  07               4.0        RLC                     ;POSITION BIT FOR COMPARE
1770  449E  4F               4.0        MOV     C,A             ;SAVE IN REG. C
1771  449F  3A  BB  46       13.0        LDA     INDATP          ;GET THE EXPECTED PARITY BIT
1772  44A2  A7               4.0        ANA     A               ;SET THE CONDITION BITS
1773  44A3  CA  BE  44       10.0        JZ      TST01QO         ;GO CHECK FOR PARITY-0
1774  44A6  79               4.0        MOV     A,C             ;ELSE CHECK FOR PARITY=1
1775  44A7  FE  01           7.0        CPI     @1
1776  44A9  CA  B5  44       10.0        JZ      TST01RG
1777  44AC  78               4.0        MOV     A,B
1778  44AD                               ROUT    R05L
 (1)  44AD  D3  8A          10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  44AF  7F               4.0                MOV     A,A             ;RETRY LINK
1779  44B0                                 ERR     TST01C,TST01RG,@1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44B0  CD  09  28      18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B3  05                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B4  01                                      .BYTE   @1
 (1)  44B5  CD  15  28      18.0                TST01RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (*)  44B8  DA  76  43      10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1780                                 ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1781                                 ;<
1782  44BB  C3  D2  44       10.0        JMP     TST01SZ         ;CONTINUE WITH TEST
1783  44BE  79               4.0 TST01QO: MOV    A,C             ;CHECK FOR PARITY = ZERO
1784  44BF  A7               4.0        ANA     A               ;SET CONDITION BITS
1785  44C0  CA  CC  44       10.0        JZ      TST01TP         ;CONTINUE IF ZERO
1786  44C3  78               4.0        MOV     A,B
1787  44C4                               ROUT    R05L
 (1)  44C4  D3  8A          10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  44C6  7F               4.0                MOV     A,A             ;RETRY LINK
```

```
1789  44C7                                        ERR     TST01C,TST01TP,@1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44C7  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CA  06                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44CB  01                                            .BYTE   @1
 (1)  44CC  CD  15  28        18.0                TST01TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44CF  DA  76  43        10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1790                                        ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1791                                        ;<
1792  44D2  7A                    4.0   TST01SZ: MOV     A,D             ;GET THE CHARACTER COUNT
1793  44D3  3C                    4.0           INR     A               ;INCREMENT
1794  44D4  FE  08                7.0           CPI     $8              ;DONE?
1795  44D6  CA  DD  44           10.0           JZ      TST01N          ;YES-GO CHECK ECC
1796  44D9  57                    4.0           MOV     D,A             ;NO-CONTINUE
1797  44DA  C3  67  44           10.0           JMP     TST01Y
1798
1799  44DD  D3  0C               10.0   TST01N: OUT     RINST           ;SINGLE STEP THE READ PATH
1800  44DF  3A  80  46           13.0           LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1801  44E2                                      ROUT    EDATA           ;SAVE IN CAS
 (1)  44E2  D3  95               10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44E4  7F                    4.0                   MOV     A,A             ;RETRY LINK
1802  44E5  47                    4.0           MOV     B,A             ;SAVE IN REGISTER B
1803  44E6  DB  19               10.0           IN      ECCCOR
1804  44E8                                      ROUT    ADATA
 (1)  44E8  D3  94               10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44EA  7F                    4.0                   MOV     A,A             ;RETRY LINK
1805  44EB  B8                    4.0           CMP     B
1806  44EC  CA  FA  44           10.0           JZ      TST01U
1807  44EF  3A  BA  46           13.0           LDA     INDAT           ;GET THE DATA/COUNT
1808  44F2                                      ROUT    R05L            ;SAVE IN THE CAS
 (1)  44F2  D3  8A               10.0                   OUT     R05L            ;WRITE AC INTO R05L
 (1)  44F4  7F                    4.0                   MOV     A,A             ;RETRY LINK
1809  44F5                                      ERR     TST01C,TST01U,1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F5  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F8  07                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F9  01                                            .BYTE   1
 (1)  44FA  CD  15  28        18.0                TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FD  DA  76  43        10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1810                                        ;>CORRECTED ECC CHARACTER INCORRECT
1811                                        ;<
```

```
1813  4500  DB  1A          10.0          IN      ECCSTA          ;GET THE ECC STATUS
1814  4502  E6  0F           7.0          ANI     @17             ;REMOVE UNWANTED BITS
1815  4504  FE  09           7.0          CPI     E.STEC!E.PNTR
1816  4506  CA  1C  45      10.0          JZ      TST01V
1817  4509                               ROUT    ADATA
 (1)  4509  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  450B  7F           4.0                      MOV     A,A             ;RETRY LINK
1818  450C  3E  09           7.0          MVI     A,E.STEC!E.PNTR
1819  450E                               ROUT    EDATA
 (1)  450E  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4510  7F           4.0                      MOV     A,A             ;RETRY LINK
1820  4511  3A  BA  46      13.0          LDA     INDAT           ;GET THE DATA/COUNT
1821  4514                               ROUT    R05L            ;SAVE IN THE CAS
 (1)  4514  D3  8A          10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  4516  7F           4.0                      MOV     A,A             ;RETRY LINK
1822  4517                               ERRB    TST01C,TST01V,1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4517  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  451A  08                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  451B  01                                  .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  451C  CD  15  28      18.0          TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  451F  DA  76  43      10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1823                                   ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1824                                   ;<
1825  4522  3A  BA  46      13.0          LDA     INDAT
1826  4525  3C           4.0          INR     A
1827  4526  32  BA  46      13.0          STA     INDAT
1828  4529  C2  76  43      10.0          JNZ     TST01C
1829  452C                               ENDTST  TST01X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  452C                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  452C  CD  06  28      18.0                  CALL    REQST
 (2)  452F  00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  4530  00  00                              .WORD                   ;SYSTEM '#' COUNT
 (2)  4532  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4534  00                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4535  07                                  .BYTE   7               ;REQUEST CODE
 (1)  4536  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4539  3D           4.0          DCR     A               ;DOWNCOUNT
 (1)  453A  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  453D  F2  4E  43      10.0          JP      TST01X          ;DO TEST UNTIL TILL = 0
1830  4540  C3  18  28      10.0  EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*************************************************************************
4543                            12.0    CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4544  C5                        12.0            PUSH    B
4545  06  00                     7.0            MVI     B,0             ;START TO CLEAR AT PORT #0
4547  DB  E0                    10.0    CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4549  E6  80                     7.0            ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
454B  B0                         4.0            ORA     B               ;ADD IN THE SELECTED PORT #
454C  D3  E0                    10.0            OUT     MBSEL           ;RESET TO THIS PORT
454E  3E  80                     7.0            MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4550  D3  40                    10.0            OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4552  AF                         4.0            XRA     A               ;CLEAR TU COMMAND A
4553  D3  40                    10.0            OUT     TCMD            ;
4555  3E  81                     7.0            MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4557  D3  40                    10.0            OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4559  3E  00                     7.0            MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
455B  D3  40                    10.0            OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
455D  AF                         4.0            XRA     A
455E  D3  44                    10.0            OUT     TAMT            ;CLEAR AMTIE WORD
4560  D3  48                    10.0            OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
4562  D3  4C                    10.0            OUT     PENAB           ;CLEAR PORT ENABLE WORD
4564  04                         4.0            INR     B               ;POINT TO THE NEXT PORT TO CLEAR
4565  78                         4.0            MOV     A,B
4566  FE  04                     7.0            CPI     4               ;DONE?
4568  C2  47  45                10.0            JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
456B  C1                        10.0            POP     B               ;RESET B & C
456C  F1                        10.0            POP     PSW             ;ALL DONE
456D  C9                        10.0            RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
                                SSUB
                                ;*********************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
                                SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE, AT THE ENTRY
                                ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                ;*BUT NOT CORRECT FOR THE INPUT DATA.
                                SP
                                ;*********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    CLEAR THE BAD TRANSLATION POINTERS
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 1
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    INCREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 10
                                ;*    ENDDO
                                ;*ENDSUB
                                S
                                ;*********************************************************************************

                                
456E  F5                12.0  BADT4X5: PUSH   PSW             ;SAVE REGISTER A + PSW
456F  C3  7D  45        10.0           JMP    BAD
4572  F5                12.0  T4X5:    PUSH   PSW             ;SAVE REGISTER A + PSW
4573  AF                 4.0           XRA    A               ;CLEAR A
4574  32  55  46        13.0           STA    BAD1            ;CLEAR BAD TRANSLATION 1
4577  32  56  46        13.0           STA    BAD2            ;CLEAR BAD TRANSLATION 2
457A  32  57  46        13.0           STA    BAD3            ;CLEAR BAD TRANSLATION 3
457D  C5                12.0  BAD:     PUSH   B               ;SAVE REGISTER B + C
457E  D5                12.0           PUSH   D               ;SAVE REGISTER D + E
457F  E5                12.0           PUSH   H               ;SAVE REGISTER H + L
```

```
1921                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
1922                                        ;CLEAR THE TRANSLATED DATA TABLE.
1923  4580  06  0A          7.0           MVI     B,10            ;SET UP LOOP COUNT
1924  4582  11  49  46     10.0           LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
1925  4585  AF              4.0           XRA     A               ;CLEAR A
1926  4586  12              7.0  D4X5:    STAX    D               ;STORE IN THE TABLE
1927  4587  13              6.0           INX     D               ;UPDATE TABLE POINTER
1928  4588  05              4.0           DCR     B               ;DECREMENT LOOP COUNT
1929  4589  C2  86  45     10.0           JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
1930  458C  0E  01          7.0           MVI     C,1             ;SET UP TRACK COUNT
1931  458E  06  04          7.0  B4X5:    MVI     B,4             ;SET UP BIT COUNT
1932  4590  11  45  46     10.0           LXI     D,TRNIN
1933  4593  AF              4.0           XRA     A               ;CLEAR THE GROUP POSITION COUNT
1934  4594  32  53  46     13.0           STA     GP4X5
1935  4597  1A              7.0  C4X5:    LDAX    D               ;GET A DATA BYTE
1936  4598  A7              4.0           ANA     A               ;SET CONDITION BITS
1937  4599  E2  9D  45     10.0           JPO     PO4X5           ;ODD PARITY LEAVE CARRY CLEAR
1938  459C  37              4.0           STC                     ;EVEN PARITY SET CARRY
1939  459D  1F              4.0  PO4X5:   RAR                     ;SHIFT OUT DESIRED BIT
1940  459E  12              7.0           STAX    D               ;STORE RESULT BACK IN TEMP TABLE
1941  459F  3A  53  46     13.0           LDA     GP4X5           ;GET THE GROUP POSITION BYTE
1942  45A2  17              4.0           RAL                     ;PUT IN THIS DATA BIT
1943  45A3  32  53  46     13.0           STA     GP4X5           ;SAVE THE UPDATED GROUP POSITION BYTE
1944  45A6  13              6.0           INX     D               ;UPDATE THE TABLE POINTER
1945  45A7  05              4.0           DCR     B               ;DECREMENT THE BIT COUNT
1946  45A8  C2  97  45     10.0           JNZ     C4X5            ;DO UNTIL ALL 4 BITS ARE RECEIVED
1947
1948                                        ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTE
1949
1950                                        ;PERFORM A DOUBLE PRECISION SHIFT RIGHT ON THE TRANSLATED DATA
1951  45AB  06  05          7.0           MVI     B,5             ;SET UP LOOP COUNT
1952  45AD  11  49  46     10.0           LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
1953  45B0  13              6.0  E4X5:    INX     D               ;POINT TO PARITY BIT
1954  45B1  1A              7.0           LDAX    D               ;GET PARITY BIT
1955  45B2  A7              4.0           ANA     A               ;CLEAR THE CARRY BIT
1956  45B3  1F              4.0           RAR                     ;SHIFT TO THE CARRY BIT
1957  45B4  12              7.0           STAX    D               ;STORE IT BACK
1958  45B5  1B              6.0           DCX     D               ;DECREMENT TO DATA BITS
1959  45B6  1A              7.0           LDAX    D               ;GET THE DATA BITS
1960  45B7  1F              4.0           RAR                     ;SHIFT IN CARRY BITS
1961  45B8  12              7.0           STAX    D               ;STORE BACK IN TABLE
1962  45B9  13              6.0           INX     D               ;POINT TO NEXT TABLE ENTRY
1963  45BA  13              6.0           INX     D               ;
1964  45BB  05              4.0           DCR     B               ;DECREMENT LOOP COUNT
1965  45BC  C2  B0  45     10.0           JNZ     E4X5            ;DO UNTIL LOOP COUNT=0
1966  45BF  3A  55  46     13.0           LDA     BAD1            ;IS A BAD POINTER SPECIFIED?
1967  45C2  A7              4.0           ANA     A               ;SET CONDITION BITS
1968  45C3  CA  CA  45     10.0           JZ      BAD1C           ;NO-CONTINUE
1969  45C6  B9              4.0           CMP     C               ;YES-IS IT THIS DATA POSITION?
1970  45C7  CA  E0  45     10.0           JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
1971  45CA  3A  56  46     13.0  BAD1C:   LDA     BAD2            ;IS A BAD POINTER SPECIFIED
1972  45CD  A7              4.0           ANA     A               ;SET CONDITION BITS
1973  45CE  CA  D5  45     10.0           JZ      BAD2C           ;NO-CONTINUE
1974  45D1  B9              4.0           CMP     C               ;YES-IS IT THIS DATA POSITION
```

```
1975  45D2  CA  E0  45   10.0          JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
1976  45D5  3A  57  46   13.0  BAD2C:  LDA     BAD3            ;IS A BAD POINTER SPECIFIED
1977  45D8  A7               4.0          ANA     A               ;SET CONDITION BITS
1978  45D9  CA  E7  45   10.0          JZ      BAD3C           ;NO-PERFORM PROPER TRANSLATION
1979  45DC  B9               4.0          CMP     C               ;YES-IS IT THIS DATA POSITION
1980  45DD  C2  E7  45   10.0          JNZ     BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
1981  45E0  3A  53  46   13.0  INV4X5: LDA     GP4X5           ;GET THE GROUP POSITION COUNT
1982  45E3  3C               4.0          INR     A               ;INCREMENT-TO INCORRECT TRANSLATION
1983  45E4  32  53  46   13.0          STA     GP4X5           ;STORE IT
1984  45E7  11  35  46   10.0  BAD3C:  LXI     D,TAB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
1985  45EA  26  00            7.0          MVI     H,0             ;CLEAR REGISTER H
1986  45EC  3A  53  46   13.0          LDA     GP4X5           ;GET GROUP POSITION COUNT
1987  45EF  E6  0F            7.0          ANI     $0F
1988  45F1  6F               4.0          MOV     L,A
1989  45F2  19              10.0          DAD     D               ;ADD GROUP POSITION COUNT TO TABLE
1990  45F3  7E               7.0          MOV     A,M             ;GET TRANSLATED DATA
1991  45F4  32  54  46   13.0          STA     TRNTMP
1992                                ;''TRNTMP'' = THE TRANSLATED DATA
1993
1994  45F7  11  49  46   10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
1995  45FA  06  05            7.0          MVI     B,5             ;SET UP LOOP COUNT
1996  45FC  3A  54  46   13.0  F4X5:   LDA     TRNTMP          ;GET TRANSLATED DATA
1997  45FF  13               6.0          INX     D               ;POINT TO PARITY POSITION
1998  4600  A7               4.0          ANA     A               ;CLEAR CARRY BIT
1999  4601  1F               4.0          RAR                     ;SHIFT OUT A BIT
2000  4602  32  54  46   13.0          STA     TRNTMP          ;STORE TRANSLATED DATA
2001  4605  1A               7.0          LDAX    D               ;GET THE PARITY BIT (ALWAYS ZERO)
2002  4606  17               4.0          RAL                     ;ROLL IN THE CARRY BIT
2003  4607  12               7.0          STAX    D               ;STORE AS THE PARITY BIT
2004  4608  13               6.0          INX     D               ;UPDATE TABLE POINTER
2005  4609  05               4.0          DCR     B               ;DECREMENT THE LOOP COUNT
2006  460A  C2  FC  45   10.0          JNZ     F4X5            ;DO UNTIL LOOP COUNT=0
2007
2008                                ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2009
2010  460D  0C               4.0          INR     C               ;DECREMENT THE TRACK COUNT
2011  460E  79               4.0          MOV     A,C
2012  460F  FE  0A            7.0          CPI     10
2013  4611  C2  8E  45   10.0          JNZ     B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2014  4614  E1              10.0          POP     H               ;RESTORE REGISTER H + L
2015  4615  D1              10.0          POP     D               ;RESTORE REGISTER D + E
2016  4616  C1              10.0          POP     B               ;RESTORE REGISTER B + C
2017  4617  F1              10.0          POP     PSW             ;RESTORE REGISTER A + PSW
2018  4618  C9              10.0          RET                     ;RETURN TO USER
2019
```

```
2021                                    .SBTTL  SUBROUTINE CALCULATE POINTER
2022                                    ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
2023                                    ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
2024                                    ;*
2025                                    ;*      INPUT/EXPECTED
2026                                    ;*          DATA                        POINTER GENERATED
2027                                    ;*      --------------                  -----------------
2028                                    ;*
2029                                    ;*         XX00XXXX                     HISTORY
2030                                    ;*         XX01XXXX                     PHASE TIE
2031                                    ;*         XX10XXXX                     AMTIE
2032                                    ;*         XX11XXXX                     ILLEGAL 5X4
2033                                    ;*
2034                                    ;*      X = DON'T CARE BITS
2035  4619                              SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2036
2037  4619  F5          12.0  POINTER: PUSH   PSW  !          ;SAVE THE ACCUMULATOR
2038  461A  C5          12.0          PUSH   B                ;SAVE B + C
2039  461B  3A  BA  46  13.0          LDA    INDAT            ;GET INPUT DATA
2040  461E  E6  30       7.0          ANI    $30              ;REMOVE THE WORKING BITS
2041  4620  07           4.0          RLC                     ;POSITION THE DATA
2042  4621  07           4.0          RLC
2043  4622  07           4.0          RLC
2044  4623  07           4.0          RLC
2045  4624  21  31  46  10.0          LXI    H,POINTBL        ;GET BASE ADDRESS OF TABLE
2046  4627  4F           4.0          MOV    C,A
2047  4628  06  00       7.0          MVI    B,0
2048  462A  09          10.0          DAD    B
2049  462B  7E           7.0          MOV    A,M              ;GET THE ECC POINTER
2050  462C  D3  0A      10.0          OUT    RTIEB            ;SET THE TIE BUS VALUE
2051  462E  C1          10.0          POP    B                ;RESTORE B + C
2052  462F  F1          10.0          POP    PSW              ;RESTORE ACCUMULATOR
2053  4630  C9          10.0          RET
2054
2055  4631  01                POINTBL: .BYTE  1
2056  4632  02                         .BYTE  2
2057  4633  04                         .BYTE  4
2058  4634  08                         .BYTE  8
2059
```

```
                                .SBTTL  TABLE 4 X 5 TRANSLATION

                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
                                ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
                                ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
                                ;
                                ;        0000                              10011
                                ;        0001                              11011
                                ;        0010                              01001
                                ;        0011                              11001
                                ;        0100                              10111
                                ;        0101                              10101
                                ;        0110                              01101
                                ;        0111                              11101
                                ;        1000                              01011
                                ;        1001                              10010
                                ;        1010                              01010
                                ;        1011                              11010
                                ;        1100                              01111
                                ;        1101                              10110
                                ;        1110                              01110
                                ;        1111                              11110

4635  13                        TAB4X5: .BYTE   @23
4636  1B                                .BYTE   @33
4637  09                                .BYTE   @11
4638  19                                .BYTE   @31
4639  17                                .BYTE   @27
463A  15                                .BYTE   @25
463B  0D                                .BYTE   @15
463C  1D                                .BYTE   @35
463D  0B                                .BYTE   @13
463E  12                                .BYTE   @22
463F  0A                                .BYTE   @12
4640  1A                                .BYTE   @32
4641  0F                                .BYTE   @17
4642  16                                .BYTE   @26
4643  0E                                .BYTE   @16
4644  1E                                .BYTE   @36
```

```
                                        .SBTTL  SUBROUTINE VARIABLES

                                TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
                                                                ;BE TRANSLATED
                                TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
                                                                ;TRANSLATION
                                GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
                                                                ;BE TRANSLATED
                                TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
                                BAD1:   .BYTE   0               ;BAD TRANSLATION POINTER 1
                                BAD2:   .BYTE   0               ;BAD TRANSLATION POINTER 2
                                BAD3:   .BYTE   0               ;BAD TRANSLATION POINTER 3
```

```
2116                                    .SBTTL  SUBROUTINE CLEAR ECC
2117  4658                              $
 (1)                                    ; ************************************************************************
2118                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
2119                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2120                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2121  4658                              $
 (1)                                    ; ************************************************************************
2122
2123  4658  AF                  4.0   CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
2124  4659  32  80  46           13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
2125  465C  C9                   10.0          RET                     ;RETURN TO USER
2126
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        ;
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        ;
                                        ; ************************************************************************

                                ECC:    LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
                                        XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
                                        MOV     E,A             ;SAVE XOR RESULT IN E
                                        ANI     $10             ;IS BIT #4 OF RESULT SET
                                        MOV     A,E             ;RESTORE XOR RESULT FROM B
                                        JZ      ECC1            ;CONTINUE IF BIT #4 RESET
                                        XRI     $23             ;ELSE-XOR WITH 23
                                ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                        ;
                                        XRA     A               ;CLEAR A
                                        MOV     C,A             ;CLEAR THE TRANSLATE RESULT
                                        LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
                                        CALL    TRANS           ;TRANSLATE THE BITS
                                        MOV     A,C             ;GET THE TRANSLATED RESULT
                                        STA     ECCCHR          ;STORE RESULT
                                        RET

                                ECCTBL: $08                     ;BIT 0   POSITION 3
                                        $20                     ;BIT 1 = POSITION 5
                                        $02                     ;BIT 2 = POSITION 1
                                        $40                     ;BIT 3 = POSITION 6
                                        $80                     ;BIT 4 = POSITION 7
                                        $01                     ;BIT 5 = POSITION 0
                                        $10                     ;BIT 6 = POSITION 4
                                        $04                     ;BIT 7 = POSITION 2

                                ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

4681  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
4683  78               4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
4684  A0               4.0          ANA   B          ;SEE IF BIT POSITION IN 'B' IS SET
4685  CA  88  46      10.0          JZ    TRANS2     ;DO NEXT BIT POSITION IF NOT SET
4688  79               4.0          MOV   A,C        ;GET PREVIOUS RESULT OF 'OR'
4689  B6               7.0          ORA   M          ;'OR' IN NEW POSITION
468A  4F               4.0          MOV   C,A        ;SAVE RESULT

468B  78               4.0  TRANS2: MOV   A,B
468C  07               4.0          RLC              ;POSITION MASK TO NEXT BIT
468D  47               4.0          MOV   B,A
468E  D8              12.0          RC               ;EXIT WHEN ALL POSITIONS DONE
468F  23               6.0          INX   H          ;POINT TO NEXT TABLE ENTRY
4690  C3  83  46      10.0          JMP   TRANS1     ;PROCESS NEXT BIT
```

```
                                .SBTTL  SUBROUTINE TRK
                                SSUB
                                ;*********************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*GENERATE BAD TRACK TRANSLATION
                                SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE GENRATES A TRACK NUMBER FOR THE BADT4X5 SUBROUTINE
                                ;*BASED ON THE VALUE OF INDAT AND AN INCORRECT TIE BUS TIME (THE TIME FOR THE NEXT BIT).
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITION       BIT POSITION OF
                                ;*          DATA                WITH ECC CORRECTION       INCORRECT POINTER
                                ;*      --------------          -------------------     --------------
                                ;*
                                ;*          XXXX0000                    0                       1
                                ;*          XXXX0001                    1                       2
                                ;*          XXXX0010                    2                       3
                                ;*          XXXX0011                    3                       4
                                ;*          XXXX0100                    4                       5
                                ;*          XXXX0101                    5                       6
                                ;*          XXXX0110                    6                       7
                                ;*          XXXX0111                    7                       P
                                ;*          XXXX1000                    P                       0
                                ;*          XXXX1001                    1                       2
                                ;*          XXXX1010                    2                       3
                                ;*          XXXX1011                    3                       4
                                ;*          XXXX1100                    4                       5
                                ;*          XXXX1101                    5                       6
                                ;*          XXXX1110                    6                       7
                                ;*          XXXX1111                    7                       P
                                ;*
                                ;*      X = DON'T CARE BITS
                                ;*
                                
4693  3A BA 46        13.0  TRK:    LDA     INDAT           ;GET INPUT DATA
4696  E6 0F            7.0         ANI     $0F             ;REMOVE 4 BITS
4698  FE 08            7.0         CPI     $8              ;PARITY TRACK?
469A  CA 9F 46        10.0         JZ      TRKA            ;YES
469D  E6 07            7.0         ANI     $7              ;NO-IGNORE BIT3
469F  3C               4.0  TRKA:   INR     A               ; INCREMENT FOR SUBROUTINE
46A0  32 55 46        13.0         STA     BAD1            ;SAVE IT.
46A3  3D               4.0         DCR     A               ;PREPARE TO USE AS TABLE INDEX
46A4  21 B0 46        10.0         LXI     H,TIMTBL        ;INCORRECT TIE BUS TIME TABLE START
46A7  5F               4.0         MOV     E,A             ;LOW HALF OF TABLE INDEX
46A8  16 00            7.0         MVI     D,0             ;ZERO HIGH HALF
46AA  19              10.0         DAD     D               ;PRODUCE TABLE ENTRY ADDRESS
46AB  7E               7.0         MOV     A,M             ;GET INCORRECT TIE BUS TIME FROM TABLE
```

```
2235  46AC  32  B9  46        13.0          STA     TIETIM          ;SAVE VALUE
2236  46AF  C9                10.0          RET
2237  46B0  03                        TIMTBL: .BYTE   3               ;INCORRECT TIE BUS TIME FOR BIT 0
2238  46B1  00                                .BYTE   0               ;INCORRECT TIE BUS TIME FOR BIT 1
2239  46B2  01                                .BYTE   1               ;INCORRECT TIE BUS TIME FOR BIT 2
2240  46B3  04                                .BYTE   4               ;INCORRECT TIE BUS TIME FOR BIT 3
2241  46B4  08                                .BYTE   8               ;INCORRECT TIE BUS TIME FOR BIT 4
2242  46B5  07                                .BYTE   7               ;INCORRECT TIE BUS TIME FOR BIT 5
2243  46B6  05                                .BYTE   5               ;INCORRECT TIE BUS TIMF FOR BIT 6
2244  46B7  06                                .BYTE   6               ;INCORRECT TIE BUS TIME FOR BIT 7
2245  46B8  02                                .BYTE   2               ;INCORRECT TIE BUS TIME FOR BIT P
2246
2247
```

```
2249                                    .SBTTL  PROGRAM VARIABLES
2250
2251  46B9  00                          TIETIM: .BYTE   0               ;TIE BUS JAM TIME
2252  46BA  00                          INDAT:  .BYTE   0               ;DATA BYTE TO ECC
2253  46BB  00                          INDATP: .BYTE   0               ;DATA BYTE PARITY
2254  46BC  00                          UNITMP: .BYTE   0               ;UNIT MAP
2255  46BD  00  00                      AMTMSK  .WORD   0               ;AMTIE MASK WORD
2256  46BF  00                          DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2257  46C0  00                          DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2258  46C1  00                          DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2259  46C2  00                          DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2260        0000                                .END
```

```
A      =%0007      ADATA  = 0094      AMTIEP= 0001      AMTIE7= 0002
AMTMSK  46BD       ARAIDF= 0098      ASAVE   4F9B      ATTCD   4F97
AXNUM   4F91       B      =%0000      BAD     457D      BADST  = 0090
BADT4X  456E       BAD1    4655       BAD1C   45CA      BAD2    4656
BAD2C   45D5       BAD3    4657       BAD3C   45E7      BIT0   = 0001
BIT1   = 0002      BIT15  = 8000      BIT2   = 0004      BIT3   = 0008
BIT4   = 0010      BIT5   = 0020      BIT6   = 0040      BIT7   = 0080
BIT8   = 0100      BIT9   = 0200      BRKPBC= 4F0A      BRKRAM= 4F10
BRKSTR= 4E60       BRKXCT= 4F00      BSAVE   4F9C      BYTCNT= 00D4
BYTEH   4F24       BYTFL   4F23      B4X5    458E      C      =%0001
CASCT   4F21       CASCTL= 00A0      CASSTA= 00A0      CATTH  = 0089
CATTL  = 0088      CBUSST= 00A1      CBYTH  = 008B      CBYTL  = 008A
CDG1H  = 0087      CDG1L  = 0086      CDG2H  = 0093      CDG2L  = 0092
CDG3H  = 0095      CDG3L  = 0094      CDVTH  = 008D      CDVTL  = 008C
CHPTIE= 0028       CHOTIE= 0020      CH1TIE= 0021      CH2TIE= 0022
CH3TIE= 0023       CH4TIE= 0024      CH5TIE= 0025      CH6TIE= 0026
CH7TIE= 0027       CKLOP  = 2815      CLEAR   4543      CLECC   4658
CLKCTL= 00F0       CLOCK   4F26      CLRLP   4547      CMC0H  = 0099
CMC0L  = 0098      CMC1H  = 009B      CMC1L  = 009A      CMC2H  = 009D
CMC2L  = C09C      CMC3H  = 009F      CMC3L  = 009E      CMINH  = 0097
CMINL  = 0096      CNTCTL= 00D7      CRCWRD= 0018      CSAVE   4F9D
CSRLH  = 0091      CSRLL  = 0090      CTCH   = 0085      CTCL   = 0084
CTSTH  = 008F      CTSTL  = 008E      CXCTH  = 0081      CXCTL  = 0080
CXINH  = 0083      CXINL  = 0082      C.     = 0001      C.AVAI= 0080
C.DP   = 0008      C.DSE  = 0010      C.DTU  = 0003      C.DVA  = 0008
C.FAIL= 00FC       C.FMT  = 0070      C.FNCT= 003E      C.GO   = 0001
C.INTC= 00FE       C.MAIN= 0020      C.NSA  = 0080      C.RCT  = 00FC
C.SER  = 0080      C.SHR  = 0040      C.SKPC= 000F      C.TAPE= 0040
C.WCS  = 0002      C4X5    4597      D      =%0002      DATAA   46BF
DATAAP  46C0       DATACT= 00D0      DATAE   46C1      DATAEP  46C2
DBUS    4F28       DBUSCT= 00C0      DBUSST= 00C0      DDRA   = 00D8
DDRAIN= 0010       DDRB   = 00D9      DDRBIN= 0002      DDRC   = 00DA
DDRCIN= 0001       DDRCO  = 0088      DDRCTL= 00DB      DIAFLG  4F22
DIAGPG= 4300       DIAGRM= 4F90      DIARD  = 000B      DONE1  = 0045
DONINT= 0010       DSAVE   4F9E      DSE    = 0006      DUMMU   4416  G
DUMMV   445C  G    DUMMY   431E  G   D.ATH0= 0001      D.ATH1= 0002
D.EOTD= 0010       D.LAGC= 0020      D.NOTW= 0040      D.NTHR= 0004
D.TACH= 0008       D.WR4  = 0080      D4X5    4586      E      =%0003
ECC     465D       ECCBAD= 0042      ECCCHR  4680      ECCCOR= 0019
ECCOK  = 0041      ECCSTA= 001A      ECCTBL  4678      ECCTST= 000E
ECC1    466A       EDATA  = 0095      EOTCLR= 0003      ERFLG   4F93
ERLP   = 2809      ERLPA  = 280F      ERLPB  = 2812      ERLPE  = 280C
ERNUM   4F90       ERRCNT= 00D6      ESAVE   4F9F      EXIT    4540
E.ACRC= 0010       E.AMT  = 0020      E.CDP  = 0080      E.CRC  = 0080
E.PNTR= 0008       E.RPE  = 0040      E.STEC= 0001      E.TTEC= 0002
E.UNC  - 0004      E4X5    45B0      FIFORD= 006A      FORMAT  4F25
FOUND   4344       FWDTST= 0061      F4X5    45FC      GCRID  = 0089
GCRSET= 0002       GOODTM= 0092      GP4X5   4653      H      =%0004
HLSAVE  4FA0       IE     = 0008      INDAT   46BA      INDATP  46BB
INTSTA= 00E0       INV4X5  45E0      ITERA   4F9A      I.PWR  = 0020
I.RMPE= 0040       I5.5   = 0010      I6.5   = 0020      I7.5   = 0040
KCALL  = 005F      KCLR   = 007B      KDEP   = 003F      KENAB  = 0078
KEXAM  = 003E      KEYBRD= 00C8      KEY1   = 0078      KEY10  = 006D
KEY11  = 006E      KEY12  = 006F      KEY13  - 005C      KEY14  = 005D
```

```
KEY15 = 005E    KEY16 = 005F    KEY17 = 003C    KEY18 = 003D
KEY19 = 003E    KEY2  = 0079    KEY20 = 003F    KEY3  = 007A
KEY4  = 007B    KEY5  = 0074    KEY6  = 0075    KEY7  = 0076
KEY8  = 0077    KEY9  = 006C    KINTA = 006F    KLDAD = 003D
KN0   = 003C    KN1   = 005C    KN2   = 005D    KN3   = 005E
KN4   = 006C    KN5   = 006D    KN6   = 006E    KN7   = 0074
KN8   = 0075    KN9   = 0076    KU2   = 0079    KU3   = 007A
KU8   = 0077    L     =%0005    LBLANK= 000F    LCE   = 0008
LCH   = 000C    LCL   = 000D    LCP   = 000E    LC0   = 0000
LC1   = 0001    LC2   = 0002    LC3   = 0003    LC4   = 0004
LC5   = 0005    LC6   = 0006    LC7   = 0007    LC8   = 0008
LC9   = 0009    LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC
LDLEDD= 00CD    LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800
LKKBD = 004C    LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055
LKLWPG= 0052    LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046
LPFLG   4F94    LPNUM   4F92    M     =%0006    MBSEL = 00E0
MB.A  = 0008    MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A
MM    = 8000    MSE   = 0008    MSGN  = 0008    MTACLR= 0000
MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040
MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080
MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002
MT.REV= 0020    MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080
M.CAPE= 0020    M.CONT= 0080    M.DEM = 0020    M.EBL = 0010
M.EXC = 0008    M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010
M.OCC = 0020    M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080
M.RDEN= 0002    M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001
M.TRA = 0040    M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080
M.WREN= 0080    M5.5  = 0001    M6.5  = 0002    M7.5  = 0004
NOTCAP= 0088    OKAY  = 00FF    OPRRAM= 4300    OPSTRT= 0058
OPVER = 0040    PADCNT= 00D5    PADCRC= 0080    PDIAG = 0048
PEID  = 008A    PENAB = 004C    PESET = 0001    PL    = 00B1
POINTB  4631    POINTE  4619    PO4X5   459D    PRDD  = 004C
PRENF = 009C    PS    = 00B2    PSTAT = 0048    PSW   =%0009
P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080
P.LCS = 0040    P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004
P.RPST= 0002    P.RP0E= 0020    P.RP1E= 0010    P.RP2E= 0020
P.RP3E= 0010    P.SING= 0080    P.STAT= 0002    P.STPE= 0080
P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040
P.WFLP= 0001    P.WPEN= 0010    P.WP0E= 0008    P.WP1E= 0004
P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B
QUEM  = 281E    RAMT  = 0010    RARA  = 0006    RARAI = 0004
RCHBD0= 0048    RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C
RCLRT = 000D    RCMD  = 0008    RCMLP = 0003    RCONT = 0080
RDATA = 0017    RDCLK = 0010    RDON  = 0011    READG = 0007
REND  = 0014    REQST = 2806    RESCHR= 00D1    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 000C    RMCTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
R.DATA= 0040    R.DON = 0002    R.DRDY= 0010    R.END = 0010
```

```
R.ILL  = 0004          R.JVOK= 0004          R.MK2  = 0008          R.PLOD= 0008
R.PLOO= 0010          R.PLO1= 0020          R.POST= 0020          R.STNM= 0002
R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040
R.VOK  = 0080          R00H   = 0081          R00L   = 0080          R01H   = 0083
R01L   = 0082          R02H   = 0085          R02L   = 0084          R03H   = 0087
R03L   = 0086          R04H   = 0089          R04L   = 0088          R05H   = 0088
R05L   = 008A          R06H   = 008D          R06L   = 008C          R07H   = 008F
R07L   = 008E          R10H   = 0091          R10L   = 0090          R11H   = 0093
R11L   = 0092          R12H   = 0095          R12L   = 0094          R13H   = 0097
R13L   = 0096          R14H   = 0099          R14L   = 0098          R15H   = 009B
R15L   = 009A          R16H   = 009D          R16L   = 009C          R17H   = 009F
R17L   = 009E          R7.5   = 0010          SELCLR= 0000          SETATA= 00A1
SID    = 0080          SOD    = 0080          SOE    = 0040          SP     =%0008
SSCLK  = 0040          SSTEP  = 0005          STACK  = 4FFF          STATRM= 4F20
STPCT    4F20          STRSP  = 5000          TAB4X5   4635          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT   = 0044
TASEL  = 0080          TCMD   = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP     4F99
TEST1    4300          TIETIM   46B9          TIMTBL   46B0          TMF    = 0099
TMRDY  = 0040          TRANS    4681          TRANS1   4683          TRANS2   468B
TRK      4693          TRKA     469F          TRKENA= 00D2          TRNIN    4645
TRNOUT   4649          TRNTMP   4654          TSET   = 2803          TSTEND= 2818
TSTS   = 0040          TST01A   441F          TST01B   4423          TST01C   4376
TST01F   438E          TST01G   43B7          TST01H   43E5          TST01I   439A
TST01K   4465          TST01L   447B          TST01M   4493  G       TST01N   44DD
TST01Q   44BE          TST01R   44B5  G       TST01S   44D2          TST01T   44CC  G
TST01U   44FA  G       TST01V   451C  G       TST01X   434E          TST01Y   4467
TUSEL0= 00D1          TUSEL1= 00D2          TU78   = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT  = 0004          T.EOT  = 0002          T.FPT  = 0001
T.NTHR= 0004          T.ONL  = 0020          T.PES  = 0008          T.PSBJ= 0020
T.PSOJ= 0008          T.PS1J= 0010          T.RDY  = 0080          T.RDYO= 0040
T.RWD  = 0010          T.SCLK= 0002          T4X5     4572          UIBG   = 00A1
UNITMP   46BC          UNITSL   4327          VALFC    4F98          VALTB    4F95
VELTST= 005B          WDR.P  = 0010          WRTCLK= 0000          WMCCTL= 00D3
WMCSTA= 00D0          WRTDAT= 00D3          WMCERR= 00DA          W.ACRC= 0004
W.CRC  = 0008          W.DIAG= 0002          W.DONN= 0040          W.ECC  = 0010
W.ENAB= 0080          W.ERR  = 0020          W.FMT  = 0070          W.GCR  = 0010
W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002          W.REV  = 0004
W.ROME= 0010          W.RST  = 0001          W.SKIP= 000F          W.WRIT= 0008
W.XFER= 0020          X      =%000A          X.DONN= 0080          X.ENAB= 0040
X.PEPE= 0002          X.ROME- 0001          X.WCLK= 0001          Y      =%0008
.        46C3
```

ERRORS DETECTED: 0

*ECC3.A78/PTP,ECC3-NLIST,PARAM,MACRO,LIST,ECC3
RUN-TIME: 4 5 0 SECONDS
CORE USED: 10K

```
                                .TITLE  ECC4 - ECC CONTROLLER PART #4
                                .SBTTL  TEST 1 - ECC BAD DATA TWO TRACKS CORRECT POINTERS
                                ;ID     ECC4-ERROR CORRECTION CONTROLLER PART #4
4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*ECC BAD DATA TWO TRACKS CORRECT POINTERS
4300                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM TWO
                                ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN TWO TRACKS, AND TWO
                                ;*CORRECT POINTERS ALSO PROVIDED.  THE PROGRAM CHECKS THAT THE DATA IS
                                ;*CORRECTED, AND THAT THE TWO TRACK ERROR CORRECT STATUS BIT IS SET.
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND THE DATA BIT POSITIONS OF THE INCORRECT DATA
                                ;*CAN BE DETERMINED FROM THE FOLLOWING TABLE.
                                ;*
                                ;*      BYTE COUNT AND
                                ;*      INPUT/EXPECTED          DATA BIT POSITIONS
                                ;*          DATA                WITH ECC CORRECTION
                                ;*      -------------           -------------------
                                ;*
                                ;*          XXXX0000                    0,1
                                ;*          XXXX0001                    1,2
                                ;*          XXXX0010                    2,3
                                ;*          XXXX0011                    3,4
                                ;*          XXXX0100                    4,5
                                ;*          XXXX0101                    5,6
                                ;*          XXXX0110                    6,7
                                ;*          XXXX0111                    7,P
                                ;*          XXXX1000                    P,0
                                ;*          XXXX1001                    1,2
                                ;*          XXXX1010                    2,3
                                ;*          XXXX1011                    3,4
                                ;*          XXXX1100                    4,5
                                ;*          XXXX1101                    5,6
                                ;*          XXXX1110                    6,7
                                ;*          XXXX1111                    7,P
                                ;*
                                ;*      X = DON'T CARE BITS
                                ;*
                                ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                                ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED
                                ;*          DATA                    POINTER GENERATED
                                ;*      -------------               -----------------
                                ;*
                                ;*          XX00XXXX                HISTORY
                                ;*          XX01XXXX                PHASE TIE
```

```
                        ;*          XX10XXXX                    AMTIE
                        ;*          XX11XXXX                    ILLEGAL 5X4
                        ;*
                        ;*       X = DON'T CARE BITS
1381  4300              SP
 (1)                    ; ***************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
1382                    ;*BGNTST
1383                    ;*     SET NORMAL READ PATH CLOCK
1384                    ;*     CALL SUBROUTINE CLEAR
1385                    ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1386                    ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1387                    ;*     DELAY 50 MICROSECONDS
1388                    ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1389                    ;*     CLEAR THE INPUT DATA
1390                    ;*     CLEAR BAD TRANSLATION POINTERS
1391                    ;*   BGNDO
1392                    ;*   : ENABLE THE READ PATH CLOCK
1393                    ;*   : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1394                    ;*   : CLOCK THE FIFO'S
1395                    ;*   : ISSUE CLEAR ALL COMMAND
1396                    ;*   : WAIT
1397                    ;*   : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
1398                    ;*   : CALL SUBROUTINE CLECC
1399                    ;*   : SET THE LOOP COUNT TO 7
1400                    ;*   : BGNDO
1401                    ;*   : :  GET THE INPUT DATA BYTE
1402                    ;*   : :  CALL SUBROUTINE ECC
1403                    ;*   : :  DECREMENT THE LOOP COUNT
1404                    ;*   : :  DO UNTIL THE LOOP COUNT=0
1405                    ;*   : ENDDO
1406                    ;*   : SET THE LOOP COUNT TO 3
1407                    ;*   : BGNDO
1408                    ;*   : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
1409                    ;*   : :  CALL SUBROUTINE TRK
1410                    ;*   : :  CALL SUBROUTINE BADT4X5
1411                    ;*   : :  INIT THE LOOP COUNT TO 5
1412                    ;*   : :  BGNDO
1413                    ;*   : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1414                    ;*   : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1415                    ;*   : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1416                    ;*   : :  :  CLOCK THE DATA INTO THE FIFO
1417                    ;*   : :  :  DECREMENT THE LOOP COUNT
1418                    ;*   : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1419                    ;*   : :  :  DO UNTIL LOOP COUNT = 0
1420                    ;*   : :  ENDDO
1421                    ;*   : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
1422                    ;*   : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1423                    ;*   : :  CALL SUBROUTINE TRK
1424                    ;*   : :  CALL SUBROUTINE BADT4X5
1425                    ;*   : :  INIT THE LOOP COUNT TO 5
1426                    ;*   : :  BGNDO
1427                    ;*   : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
```

```
1428                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1429                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1430                    ;*  : : :  DECREMENT THE LOOP COUNT
1431                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1432                    ;*  : : ENDDO
1433                    ;*  : : DECREMENT THE LOOP COUNT
1434                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1435                    ;*  : ENDDO
1436                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1437                    ;*  : SET UP WATCHDOG TIMER COUNT
1438                    ;*  : BGNDO
1439                    ;*  : :    SINGLE STEP THE READ PATH
1440                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1441                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1442                    ;*  : ENDDO
1443                    ;*  : IF WATCHDOG TIMER=0
1444                    ;*  : :    THEN-ERROR
1445                    ;*  : :    ELSE-CONTINUE
1446                    ;*  : ENDIF
1447                    ;*  : GET THE TIE BUS TABLE ADDRESS
1448                    ;*  : SINGLE STEP THE READ PATH
1449                    ;*  : INIT THE LOOP COUNT TO 0
1450                    ;*  : BGNDO
1451                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1 OR JAM COUNT 2
1452                    ;*  : :    :  THEN-CALL SUBROUTINE POINTER
1453                    ;*  : :    :  ELSE-CLEAR TIE BUS POINTER
1454                    ;*  : :    ENDIF
1455                    ;*  : :    SINGLE STEP THE READ PATH
1456                    ;*  : :    DECREMENT THE LOOP COUNT
1457                    ;*  : :    DO UNTIL LOOP COUNT = 9
1458                    ;*  : ENDDO
1459                    ;*  : SET UP WATCHDOG TIMER COUNT
1460                    ;*  : BGNDO
1461                    ;*  : :  SINGLE STEP THE READ PATH
1462                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1463                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1464                    ;*  : ENDDO
1465                    ;*  : IF WATCHDOG TIMER=0
1466                    ;*  : :  THEN-ERROR-EXIT TEST
1467                    ;*  : :  ELSE-CONTINUE
1468                    ;*  : ENDIF
1469                    ;*  : SINGLE STEP THE READ PATH
1470                    ;*  : INIT THE  LOOP COUNT TO 7
1471                    ;*  : BGNDO
1472                    ;*  : :  SINGLE STEP THE READ PATH
1473                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1474                    ;*  : :  IF NOT EQUAL
1475                    ;*  : :  :  THEN-ECC ERROR
1476                    ;*  : :  :  ELSE-CONTINUE
1477                    ;*  : :  ENDIF
1478                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1479                    ;*  : :  IF NOT EQUAL
1480                    ;*  : :  :  THEN-ECC ERROR
1481                    ;*  : :  :  ELSE-CONTINUE
```

```
                                ;*   : :  ENDIF
                                ;*   : :  DECREMENT THE LOOP COUNT
                                ;*   : :  DO UNTIL THE LOOP COUNT = 0
                                ;*   : ENDDO
                                ;*   : SINGLE STEP THE READ PATH
                                ;*   : IF EXPECTED NOT EQUAL
                                ;*   : :  THEN-ECC ERROR
                                ;*   : :  ELSE-CONTINUE
                                ;*   : ENDIF
                                ;*   : INPUT THE ECC STATUS
                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*   : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
                                ;*   : :  THEN-CONTINUE
                                ;*   : :  ELSE-ERROR
                                ;*   : ENDIF
                                ;*   : INCREMENT THE INPUT DATA
                                ;*   : DO UNTIL INPUT DATA = 0
                                ;*  ENDDO
                                ;*ENDTST
4300                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*ECC4 MICRO  EST 01
                                ;*ECC4 MICRO cRROR 01
                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                ;*M8951, M8950'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*ECC4 MICRO TEST 01
                                ;*ECC4 MICRO ERROR 02
                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC4 MICRO TEST 01
                                ;*ECC4 MICRO ERROR 03
                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                                ;*SENT TO THE ECC
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC4 MICRO TEST 01
                                ;*ECC4 MICRO ERROR 04
                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                ;*M8951, M8950'S
                                ;*CORRECTED DATA FROM THE ECC-INCORRECT
                                ;*BYTE/SCLK COUNT NUMBER = LLL
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 05
                                        ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 06
                                        ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 07
                                        ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 10
                                        ;*ECC4-ECC CONTROLLER-BAD DA.      TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DA'A GROUP
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        S
                                        ; ***************************************************************************
                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
                                                MVI     A,@1            ;DEFINE THE TEST NUMBER
                                                CALL    TSET            ;SETUP THE TEST
                                        ;%ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;&M8951, M8950'S
                                                REQ     @7,0,0,0,0
                                                CALL    REQST
                                                .BYTE   0               ;DATA PATTERN NUMBER
                                                .WORD   0               ;SYSTEM '0'' COUNT
                                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
                                                .BYTE   0               ;DATA COMPARE FLAG IF =1
                                                .BYTE   @7              ;REQUEST CODE
                                                RIN     R12L
                                                IN      R12L            ;READ R12L INTO AC
                                                MOV     A,A             ;RETRY LINK
                                                STA     UNITMP
                                                ANA     A
                                                JNZ     UNITSL
```

```
1576  4319                                        ERR     EXIT,DUMMY
 (1)
 (1)  4319  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                           .BYTE
 (1)  431E  CD  15  28        18.0               DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  74  47        10.0                       JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1577                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1578                                    ;>FATAL ERROR - TEST ABORTED
1579  4324  C3  74  47        10.0              JMP     EXIT
1580
1581  4327  06  00              7.0  UNITSL:  MVI     B,0
1582  4329  3A  38  49         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1583  432C  E6  01               7.0           ANI     @01             ;PORT 0?
1584  432E  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #0
1585  4331  04                   4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
1586  4332  3A  38  49         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1587  4335  E6  02               7.0           ANI     @02             ;PORT 1?
1588  4337  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #1
1589  433A  04                   4.0           INR     B               ;NO-UPDATE POINTER TO PORT #2
1590  433B  3A  38  49         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1591  433E  E6  04               7.0           ANI     @04             ;PORT 2?
1592  4340  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #2
1593  4343  04                   4.0           INR     B               ;NO-ASSUME PORT #3
1594  4344  CD  82  47         18.0  FOUND:   CALL    CLEAR           ;CLEAR ALL TU PORTS
1595  4347  DB  E0              10.0           IN      INTSTA
1596  4349  E6  80               7.0           ANI     BIT7
1597  434B  B0                   4.0           ORA     B
1598  434C  D3  E0              10.0           OUT     MBSEL
1599  434E  32  35  49         13.0           STA     SUNIT           ;SAVE FOR NEXT TEST
1600
1601  4351  3E  10               7.0  TS101X:  MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1602  4353  D3  F0              10.0           OUT     CLKCTL
1603  4355  3E  10               7.0           MVI     A,W.GCR
1604  4357  D3  D3              10.0           OUT     WMCCTL
1605  4359  3E  04               7.0           MVI     A,P.RPEN
1606  435B  D3  4C              10.0           OUT     PENAB
1607
1608  435D  3E  60               7.0           MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1609  435F  D3  48              10.0           OUT     PDIAG           ;LCS, LWR
1610  4361  AF                   4.0           XRA     A
1611  4362  D3  44              10.0           OUT     TAMT
1612  4364  3E  0A               7.0           MVI     A,10            ;DELAY
1613  4366  3D                   4.0  1$:      DCR     A               ;50
1614  4367  C2  66  43         10.0           JNZ     1$              ;MICROSECONDS
1615
1616  436A  D3  D2              10.0           OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1617  436C  D3  0A              10.0           OUT     RTIEB           ;CLEAR THE TIE BUS REGISTER
1618  436E  D3  D2              10.0           OUT     TRKENA
1619  4370  32  94  48         13.0           STA     BAD1            ;CLEAR THE INVALID
1620  4373  32  95  48         13.0           STA     BAD2            ;TRACK TRANSLATIONS
1621  4376  32  96  48         13.0           STA     BAD3            ;POINTERS
1622  4379  32  36  49         13.0           STA     INDAT           ;CLEAR THE INPUT DATA BYTE
```

```
1623  437C                                        ROUT    R05H
 (1)  437C  D3  8B          10.0                          OUT     R05H            ;WRITE AC INTO R05H
 (1)  437E  7F               4.0                          MOV     A,A             ;RETRY LINK
1624
1625  437F  3E  A8           7.0  TST01C: MVI     A,R.PLO1!R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1626                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1627  4381  D3  09          10.0          OUT     RPCTL
1628  4383  D3  08          10.0          OUT     RFIFOL          ;CLOCK THE FIFO'S
1629  4385  3E  0D           7.0          MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1630  4387  D3  0B          10.0          OUT     RCMD
1631
1632  4389  00               4.0          NOP                     ;WAIT
1633  438A  00               4.0          NOP
1634  438B  00               4.0          NOP
1635  438C  00               4.0          NOP
1636  438D  00               4.0          NOP
1637
1638  438E  3E  A9           7.0          MVI     A,R.PLO1!R.STPC!R.PLOD.R.TBJN   ;STOP THE READ PATH
1639                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1640  4390  D3  09          10.0          OUT     RPCTL
1641  4392  CD  97  48      18.0          CALL    CLECC           ;CLEAR THE ECC CHARACTER
1642  4395  16  07           7.0          MVI     D,@7            ;SET UP THE LOOP COUNT
1643  4397  3A  36  49      13.0  TST01F: LDA     INDAT           ;GET THE INPUT DATA
1644  439A  CD  9C  48      18.0          CALL    ECC             ;CALCULATE ECC FOR
1645  439D  15               4.0          DCR     D               ;THE 7 INPUT BYTES
1646  439E  C2  97  43      10.0          JNZ     TST01F
1647
1648  43A1  0E  03           7.0          MVI     C,@3            ;INIT THE LOOP COUNT
1649  43A3  3A  36  49      13.0  TST01I: LDA     INDAT           ;GET THE INPUT DATA
1650  43A6                                ROUT    R05L
 (1)  43A6  D3  8A          10.0                          OUT     R05L            ;WRITE AC INTO R05L
 (1)  43A8  7F               4.0                          MOV     A,A             ;RETRY LINK
1651  43A9  32  84  48      13.0          STA     TRNIN           ;FILL THE TRANSLATOR
1652  43AC  32  85  48      13.0          STA     TRNIN+1         ;SUBROUTINE INPUT
1653  43AF  32  86  48      13.0          STA     TRNIN+2         ;BUFFER
1654  43B2  32  87  48      13.0          STA     TRNIN+3         ;
1655  43B5  CD  D8  48      18.0          CALL    TRK
1656  43B8  AF               4.0          XRA     A               ;CLEAR THE ACCUMULATOR
1657  43B9  32  96  48      13.0          STA     BAD3            ;CLEAR THE THIRD POINTER
1658  43BC  CD  AD  47      18.0          CALL    BADT4X5         ;TRANSLATE THE SUBGROUP
1659
1660  43BF  21  88  48      10.0          LXI     H,TRNOUT        ;GET POINTER TO TRANSLATE DATA TABLE
1661  43C2  06  05           7.0          MVI     B,@5            ;SET UP LOOP COUNT
1662  43C4  7E               7.0  TST01G: MOV     A,M             ;GET A DATA BYTE
1663  43C5  D3  40          10.0          OUT     TCMD            ;STORE DATA IN COMMAND ADDRESS
1664  43C7  23               6.0          INX     H               ;POINT TO DATA PARITY
1665  43C8  7E               7.0          MOV     A,M             ;GET THE DATA PARITY
1666
1667  43C9  07               4.0          RLC                     ;POSITION FOR OUTPUT
1668  43CA  F6  60           7.0          ORI     P.LWR.P.LCS     ;OR IN CONTROL BITS
1669  43CC  D3  48          10.0          OUT     PDIAG           ;OUTPUT THE DATA PARITY
1670
1671  43CE  D3  08          10.0          OUT     RFIFOL          ;CLOCK DATA INTO THE FIFO'S
1672
```

```
1673  43D0  23              6.0          INX   H              ;UPDATE THE TABLE POINTER
1674  43D1  05              4.0          DCR   B              ;DECREMENT LOOP COUNT
1675  43D2  C2  C4  43     10.0          JNZ   TST01G         ;DO UNTIL LOOP COUNT = 0
1676
1677  43D5  3A  36  49      13.0          LDA   INDAT          ;GET THE INPUT DATA
1678  43D8  32  84  48      13.0          STA   TRNIN          ;FILL THE FIRST THREE
1679  43DB  32  85  48      13.0          STA   TRNIN+1        ;BYTES OF THE TRANSLATOR
1680  43DE  32  86  48      13.0          STA   TRNIN+2        ;SUBROUTINE WITH THE INPUT DATA
1681  43E1  3A  C5  46      13.0          LDA   ECCCHR         ;STORE THE CALCULATED
1682  43E4  32  87  48      13.0          STA   TRNIN+3        ;ECC CHARACTER AS THE LAST CHARACTER
1683  43E7  CD  D8  48      18.0          CALL  TRK
1684  43EA  AF               4.0          XRA   A              ;CLEAR THE ACCUMULATOR
1685  43EB  32  96  48      13.0          STA   BAD3           ;CLEAR THE THIRD POINTER
1686  43EE  CD  AD  47      18.0          CALL  BADT4X5        ;TRANSLATE THE SECOND SUBGROUP
1687
1688  43F1  21  88  48      10.0          LXI   H,TRNOUT       ;GET THE POINTER TO INPUT DATA TABLE
1689  43F4  06  05           7.0          MVI   B,@5           ;SET UP THE LOOP COUNT
1690  43F6  7E               7.0  TST01H: MOV   A,M            ;GET A DATA BYTE
1691  43F7  D3  40          10.0          OUT   TCMD           ;STORE DATA IN CMD ADDRESS
1692  43F9  23               6.0          INX   H              ;POINT TO DATA PARITY BIT
1693  43FA  7E               7.0          MOV   A,M            ;GET THE DATA PARITY
1694  43FB  07               4.0          RLC                  ;POSITION FOR OUTPUT
1695  43FC  F6  60           7.0          ORI   P.LWR.P.LCS    ;OR IN CONTROL BITS
1696  43FE  D3  48          10.0          OUT   PDIAG          ;OUTPUT THE DATA PARITY
1697  4400  D3  08          10.0          OUT   RFIFOL         ;CLOCK DATA INTO THE FIFOS
1698  4402  23               6.0          INX   H              ;UPDATE THE TABLE POINTER
1699  4403  05               4.0          DCR   B              ;DECREMENT THE LOOP COUNT
1700  4404  C2  F6  43      10.0          JNZ   TST01H         ;DO UNTIL LOOP COUNT=0
1701
1702  4407  0D               4.0          DCR   C              ;PUT THE GROUP IN
1703  4408  C2  A3  43      10.0          JNZ   TST01I         ;THE FIFO TWICE
1704
1705  440B  3E  0B           7.0          MVI   A,DIARD        ;LOAD THE DIAGNOSTIC READ
1706  440D  D3  0B          10.0          OUT   RCMD           ;COMMAND
1707  440F  11  01  00      10.0          LXI   D,1            ;SET WATCH DOG INCREMENT
1708  4412  21  A8  FD      10.0          LXI   H,-600         ;SET WATCH DOG COUNT TO 600
1709  4415  D3  0C          10.0  2$:     OUT   RINST          ;STEP THE READ PATH
1710  4417  DB  01          10.0          IN    RPCHI          ;DATA READY SET?
1711  4419  E6  10           7.0          ANI   R.DRDY
1712  441B  C2  30  44      10.0          JNZ   TST01A         ;YES-GO PROCESS
1713  441E  19              10.0          DAD   D              ;WATCH DOG TIMEOUT?
1714  441F  D2  15  44      10.0          JNC   2$             ;NO-CONTINUE
1715  4422                               ERR   TST01C,DUMMW,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4422  CD  09  28      18.0                CALL  ERLP           ;PROCESS ERROR - DO 2.3
 (1)       0002                         MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4425  02                         .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  4426  01                         .BYTE 1
 (1)  4427  CD  15  28      18.0  DUMMW:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  442A  DA  7F  43      10.0          JC    TST01C         ;LOOP ADDRESS IF LOOP SPECIFIED
1716                                ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1717                                ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1718                                ;<FATAL ERROR - MICRO TEST ABORTED
1719  442D  C3  74  47      10.0          JMP   EXIT
```

```
1720
1721  4430  D3  0C          10.0  TST01A: OUT   RINST
1722  4432  06  00           7.0         MVI   B,$0          ;INIT THE BIT TIME COUNT (LOOP COUNT)
1723  4434  00               4.0  TST01B: NOP
1724  4435  3A  32  49       13.0         LDA   TIET1         ;GET FIRST TIE BUS JAM TIME
1725  4438  B8               4.0         CMP   B             ;IS IT TIME TO JAM THE TIE BUS?
1726  4439  C2  42  44       10.0         JNZ   4$            ;NO - CONTINUE
1727  443C  CD  58  48       18.0         CALL  POINTER       ;YES - GO GET A REAL POINTER
1728  443F  C3  52  44       10.0         JMP   5$            ;
1729  4442  3A  33  49       13.0  4$:    LDA   TIET2         ;GET SECOND TIE BUS JAM TIME
1730  4445  B8               4.0         CMP   B             ;IS IT TIME TO JAM THE TIE BUS?
1731  4446  C2  4F  44       10.0         JNZ   6$            ;NO-CONTINUE
1732  4449  CD  58  48       18.0         CALL  POINTER       ;YES- GO GET A REAL POINTER
1733  444C  C3  52  44       10.0         JMP   5$            ;
1734  444F  AF                4.0  6$:    XRA   A             ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
1735  4450  D3  0A           10.0         OUT   RTIEB         ;LOAD THE POINTER
1736  4452  D3  0C           10.0  5$:    OUT   RINST         ;STEP CLOCK
1737  4454  04                4.0         INR   B             ;DECREMENT THE LOOP COUNT
1738  4455  78                4.0         MOV   A,B
1739  4456  FE  09            7.0         CPI   $9            ;DONE
1740  4458  C2  34  44       10.0         JNZ   TST01B        ;DO UNTIL LOOP COUNT = ZERO
1741  445B  AF                4.0         XRA   A
1742  445C  D3  0A           10.0         OUT   RTIEB         ;CLEAR THE TIE BUS REGISTER
1743  445E  D3  0C           10.0         OUT   RINST         ;CLOCK THE READ PATH
1744  4460  D3  0C           10.0         OUT   RINST         ;CLOCK THE READ PATH
1745  4462  11  01  00       10.0         LXI   D,1           ;SET WATCHDOG TIMER INCREMENT
1746  4465  21  A8  FD       10.0         LXI   H,-600        ;SET WATCHDOG COUNT TO 600
1747  4468  D3  0C           10.0  3$:    OUT   RINST         ;SINGLE STEP THE READ PATH
1748  446A  DB  01           10.0         IN    RPCHI         ;DATA READY SET?
1749  446C  E6  10            7.0         ANI   R.DRDY
1750  446E  C2  83  44       10.0         JNZ   TST01K        ;YES-GO PROCESS
1751  4471  19               10.0         DAD   D             ;NO-WATCHDOG TIMEOUT?
1752  4472  D2  68  44       10.0         JNC   3$            ;NO-CONTINUE
1753  4475                            ERR   TST01C,DUMMX,1
 (1)                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4475  CD  09  28       18.0                 CALL  ERLP          ;PROCESS ERROR - DO 2.3
 (1)       0003                                  MSGN  =     MSGN+1    ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4478  03                                    .BYTE MSGN          ;MESSAGE NUMBER ID
 (1)  4479  01                                    .BYTE 1
 (1)  447A  CD  15  28       18.0         DUMMX:: CALL  CKLOP         ;CHECK LOOP FUNCTION - DO 2.3
 (1)  447D  DA  7F  43       10.0                 JC    TST01C        ;LOOP ADDRESS IF LOOP SPECIFIED
1754                        ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1755                        ;>SENT TO THE ECC
1756                        ;<FATAL ERROR - MICRO TEST ABORTED
1757  4480  C3  74  47       10.0         JMP   EXIT
1758
1759  4483  16  01            7.0  TST01K: MVI   D,@1          ;INIT THE LOOP COUNT
1760  4485  00                4.0  TST01Y: NOP
1761  4486  3A  36  49       13.0         LDA   INDAT         ;GET THE INPUT DATA
1762  4489                            ROUT  EDATA
 (1)  4489  D3  95           10.0                 OUT   EDATA       ;WRITE AC INTO EDATA
 (1)  448B  7F                4.0                 MOV   A,A         ;RETRY LINK
1763  448C  A7                4.0         ANA   A             ;SET CONDITION BITS
1764  448D  3E  00            7.0         MVI   A,0           ;CLEAR THE ACC.
```

```
1765  448F  E2  93  44  10.0          JPO   1$                ;ODD # OF ONES - CONTINUE
1766  4492  3C              4.0      INR   A                 ;EVEN # OF ONES - SET PARITY
1767  4493  32  37  49  13.0  1$:     STA   INDATP            ;STORE THE PARITY BIT
1768  4496  D3  0C          10.0  TST01L: OUT   RINST          ;SINGLE STEP THE READ LOGIC
1769  4498  3A  36  49  13.0          LDA   INDAT
1770  449B  47              4.0      MOV   B,A
1771  449C  DB  19          10.0      IN    ECCCOR
1772  449E  B8              4.0      CMP   B
1773  449F  CA  AA  44  10.0          JZ    TST01M
1774  44A2                           ROUT  ADATA
 (1)  44A2  D3  94          10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  44A4  7F              4.0                 MOV   A,A           ;RETRY LINK
1775  44A5                         ERRB  TST01C,TST01M,@1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  44A5  CD  12  28  18.0                CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                            MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A8  04                              .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44A9  01                              .BYTE @1              ;PRINT ROUTINE NUMBER
 (1)  44AA  CD  15  28  18.0          TST01M::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44AD  DA  7F  43  10.0                JC    TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1776                           ;>CORRECTED DATA FROM THE ECC-INCORRECT
1777  44B0  DB  15          10.0      IN    RPSTA               ;GET THE ECC PARITY BIT
1778  44B2  E6  80           7.0      ANI   E.CDP               ;GET THE CORRECTED PARITY BIT
1779  44B4  07              4.0      RLC                       ;POSITION BIT FOR COMPARE
1780  44B5  4F              4.0      MOV   C,A                 ;SAVE IN REG. C
1781  44B6  3A  37  49  13.0          LDA   INDATP              ;GET THE EXPECTED PARITY BIT
1782  44B9  A7              4.0      ANA   A                   ;SET THE CONDITION BITS
1783  44BA  CA  D1  44  10.0          JZ    TST01QO             ;GO CHECK FOR PARITY-0
1784  44BD  79              4.0      MOV   A,C                 ;ELSE CHECK FOR PARITY=1
1785  44BE  FE  01           7.0      CPI   @1
1786  44C0  CA  C8  44  10.0          JZ    TST01RG
1787  44C3                           ERR   TST01C,TST01RG,@1
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44C3  CD  09  28  18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                            MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C6  05                              .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44C7  01                              .BYTE @1
 (1)  44C8  CD  15  28  18.0          TST01RG::   CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44CB  DA  7F  43  10.0                JC    TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1788                           ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1789  44CE  C3  E1  44  10.0          JMP   TST01SZ             ;CONTINUE WITH TEST
1790  44D1  79              4.0  TST01QO: MOV   A,C            ;CHECK FOR PARITY = ZERO
1791  44D2  A7              4.0      ANA   A                   ;SET CONDITION BITS
1792  44D3  CA  DB  44  10.0          JZ    TST01TP             ;CONTINUE IF ZERO
1793  44D6                           ERR   TST01C,TST01TP,@1
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44D6  CD  09  28  18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                            MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D9  06                              .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44DA  01                              .BYTE @1
 (1)  44DB  CD  15  28  18.0          TST01TP::   CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44DE  DA  7F  43  10.0                JC    TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1794                           ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1795  44E1  7A              4.0  TST01SZ: MOV   A,D              ;GET THE CHARACTER COUNT
```

```
1796  44E2  3C              4.0          INR     A               ;INCREMENT
1797  44E3  FE  08          7.0          CPI     $8              ;DONE?
1798  44E5  CA  EC  44      10.0         JZ      TST01N          ;YES-GO CHECK ECC
1799  44E8  57              4.0          MOV     D,A             ;NO-CONTINUE
1800  44E9  C3  85  44      10.0         JMP     TST01Y
1801  44EC  D3  0C          10.0  TST01N: OUT     RINST           ;SINGLE STEP THE READ PATH
1802  44EE  3A  C5  48      13.0         LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1803  44F1                               ROUT    EDATA           ;SAVE IN CAS
 (1)  44F1  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44F3  7F              4.0                  MOV     A,A             ;RETRY LINK
1804  44F4  47              4.0          MOV     B,A             ;SAVE IN REGISTER B
1805  44F5  DB  19          10.0         IN      ECCCOR
1806  44F7                               ROUT    ADATA
 (1)  44F7  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44F9  7F              4.0                  MOV     A,A             ;RETRY LINK
1807  44FA  B8              4.0          CMP     B
1808  44FB  CA  03  45      10.0         JZ      TST01U
1809  44FE                               ERR     TST01C,TST01U,1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44FE  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4501  07                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4502  01                                 .BYTE   1
 (1)  4503  CD  15  28      18.0         TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4506  DA  7F  43      10.0                 JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1810                           ;>CORRECTED ECC CHARACTER INCORRECT
1811  4509  DB  1A          10.0         IN      ECCSTA          ;GET THE ECC STATUS
1812  450B  E6  0F          7.0          ANI     @17             ;REMOVE UNWANTED BITS
1813  450D  FE  02          7.0          CPI     E.TTEC          ;IS TWO TRACK CORRECT BIT SET?
1814  450F  CA  1F  45      10.0         JZ      TST01V
1815  4512                               ROUT    ADATA
 (1)  4512  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4514  7F              4.0                  MOV     A,A             ;RETRY LINK
1816  4515  3E  02          7.0          MVI     A,E.TTEC
1817  4517                               ROUT    EDATA
 (1)  4517  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4519  7F              4.0                  MOV     A,A             ;RETRY LINK
1818  451A                               ERRB    TST01C,TST01V,1
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  451A  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  451D  08                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  451E  01                                 .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  451F  CD  15  28      18.0         TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4522  DA  7F  43      10.0                 JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1819                           ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1820  4525  3A  36  49      13.0         LDA     INDAT
1821  4528  3C              4.0          INR     A
1822  4529  32  36  49      13.0         STA     INDAT
1823  452C  C2  7F  43      10.0         JNZ     TST01C
1824  452F                               ENDTST  TST01X
 (1)                           ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  452F                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  452F  CD  06  28      18.0                 CALL    REQST
```

```
(2)  4532  00                                                .BYTE                    ;DATA PATTERN NUMBER
(2)  4533  00  00                                            .WORD                    ;SYSTEM '*' COUNT
(2)  4535  00  00                                            .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2)  4537  00                                                .BYTE                    ;DATA COMPARE FLAG IF =1
(2)  4538  07                                                .BYTE   7                ;REQUEST CODE
(1)  4539  3A  9A  4F      13.0          LDA     ITERA               ;GET ITERATION COUNT
(1)  453C  3D                4.0          DCR     A                   ;DOWNCOUNT
(1)  453D  32  9A  4F      13.0          STA     ITERA               ;SAVE COUNT
(1)  4540  F2  51  43      10.0          JP      TST01X              ;DO TEST UNTIL TILL = 0
```

```
1827
1828  4543                .SBTTL  TEST 2 - ECC GOOD DATA THREE HARD POINTERS
 (1)                      ST
 (1)                      ;*****************************************************************************
 (1)                      ;*TEST TITLE
1829                      ;*---- -----
1830  4543                ;*ECC GOOD DATA THREE HARD POINTERS
 (1)                      SD
 (1)                      ;*****************************************************************************
 (1)                      ;*DESCRIPTION
1831                      ;*-----------
1832                      ;*THIS TEST CHECKS THE ABILITY OF THE ECC TO DETECT THE UNCORRECTABLE
1833                      ;*SITUATION AND PASS THE DATA AS IS (I.E., NOT PERFORM ANY CORRECTION)
1834                      ;*TO THE CHECK CHARACTER LOGIC WITH THE UNCORRECTABLE STATUS BIT SET.
1835                      ;*THE PROGRAM LOADS CORRECT DATA TO THE FIFO AND JAMS THREE HARD POINTERS
1836                      ;*TO THE ECC BASED ON THE LAST DATA BYTE IN THE GROUP.  SEE TABLE.
1837                      ;*
1838                      ;*                                                    DATA BIT POSITION
1839                      ;*      BYTE COUNT              DATA                  WITH HARD POINTERS
1840                      ;*      ----------              ----                  ------------------
1841                      ;*
1842                      ;*         001          000,001,002,003,004,005,006        7,P,0
1843                      ;*         002          007,010,011,012,013,014,015        6,7,P
1844                      ;*         003          016,017,020,021,022,023,024        5,6,7
1845                      ;*         004          025,026,027,030,031,032,033        4,5,6
1846                      ;*         005          034,035,036,037,040,041,042        3,4,5
1847                      ;*         006          043,044,045,046,047,050,051        2,3,4
1848                      ;*         007          052,053,054,055,056,057,060        1,2,3
1849                      ;*         010          061,062,063,064,065,066,067        0,0,1
1850                      ;*         011          070,071,072,073,074,075,076        7,P,0
1851                      ;*         012          077,100,101,102,103,104,105        6,7,P
1852                      ;*         013          106,107,110,111,112,113,114        5,6,7
1853                      ;*         014          115,116,117,120,121,122,123        4,5,6
1854                      ;*         015          124,125,126,127,130,131,132        3,4,5
1855                      ;*         016          133,134,135,136,137,140,141        2,3,4
1856                      ;*         017          142,143,144,145,146,147,150        1,2,3
1857                      ;*         020          151,152,153,154,155,156,157        0,1,2
1858                      ;*         021          160,161,162,163,164,165,166        7,P,0
1859                      ;*         022          167,170,171,172,173,174,175        6,7,P
1860                      ;*         023          176,177,200,201,202,203,204        5,6,7
1861                      ;*         024          205,206,207,210,211,212,213        4,5,6
1862                      ;*         025          2   215,216,217,220,221,222        3,4,5
1863                      ;*         026          223,224,225,226,227,230,231        2,3,4
1864                      ;*         027          232,233,234,235,236,237,240        1,2,3
1865                      ;*         030          241,242,243,244,245,246,247        P,0,1
1866                      ;*         031          250,251,252,253,254,255,256        7,P,0
1867                      ;*         032          257,260,261,262,263,264,265        6,7,P
1868                      ;*         033          266,267,270,271,272,273,274        5,6,7
1869                      ;*         034          275,276,277,300,301,302,303        4,5,6
1870                      ;*         035          304,305,306,307,310,311,312        3,4,5
1871                      ;*         036          313,314,315,316,317,320,321        2,3,4
1872                      ;*         037          322,323,324,325,326,327,330        1,2,3
1873                      ;*         040          331,332,333,334,335,336,337        0,1,2
1874                      ;*         041          340,341,342,343,344,345,346        7,P,0
                          ;*         042          347,350,351,352,353,354,355        6,7,P
```

```
1875                    ;*          043          356,357,360,361,362,363,364          5,6,7
1876                    ;*          044          365,366,367,370,371,372,373          4,5,6
1877                    ;*          045          374,375,376,377,000,001,002          3,4,5
1878                    ;*
1879                    ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALWAYS BAD 4X5.
1880  4543              SP
 (1)                    ; *****************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
1881                    ;*BGNTST
1882                    ;*    SET NORMAL READ PATH CLOCK
1883                    ;*    CALL SUBROUTINE CLEAR
1884                    ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1885                    ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1886                    ;*    DELAY 50 MICROSECONDS
1887                    ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1888                    ;*    CLEAR THE INPUT DATA
1889                    ;*    CLEAR BAD TRANSLATION POINTERS
1890                    ;*  BGNDO
1891                    ;*  : ENABLE THE READ PATH CLOCK
1892                    ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1893                    ;*  : CLOCK THE FIFO'S
1894                    ;*  : ISSUE CLEAR ALL COMMAND
1895                    ;*  : WAIT
1896                    ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
1897                    ;*  : CALL SUBROUTINE CLECC
1898                    ;*  : SET THE LOOP COUNT TO 7
1899                    ;*  : BGNDO
1900                    ;*  : :  GET THE INPUT DATA BYTE
1901                    ;*  : :  STORE IN BUFFER
1902                    ;*  : :  CALL SUBROUTINE ECC
1903                    ;*  : :  DECREMENT THE LOOP COUNT
1904                    ;*  : :  INCREMENT THE INPUT DATA
1905                    ;*  : :  DO UNTIL THE LOOP COUNT=0
1906                    ;*  : ENDDO
1907                    ;*  : SET THE LOOP COUNT TO 3
1908                    ;*  : BGNDO
1909                    ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER FROM THE DATA BUFFER
1910                    ;*  : :  CALL SUBROUTINE T4X5
1911                    ;*  : :  INIT THE LOOP COUNT TO 5
1912                    ;*  : :  BGNDO
1913                    ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1914                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1915                    ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1916                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1917                    ;*  : :  :  DECREMENT THE LOOP COUNT
1918                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1919                    ;*  : :  :  DO UNTIL LOOP COUNT = 0
1920                    ;*  : :  ENDDO
1921                    ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER FROM THE DATA BUFFER
1922                    ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1923                    ;*  : :  CALL SUBROUTINE T4X5
1924                    ;*  : :  INIT THE LOOP COUNT TO 5
1925                    ;*  : :  BGNDO
```

```
1926                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1927                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1928                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1929                    ;*  : : :  DECREMENT THE LOOP COUNT
1930                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1931                    ;*  : : ENDDO
1932                    ;*  : : DECREMENT THE LOOP COUNT
1933                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1934                    ;*  : ENDDO
1935                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1936                    ;*  : SET UP WATCHDOG TIMER COUNT
1937                    ;*  : BGNDO
1938                    ;*  : :    SINGLE STEP THE READ PATH
1939                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1940                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1941                    ;*  : ENDDO
1942                    ;*  : IF WATCHDOG TIMER=0
1943                    ;*  : :    THEN-ERROR
1944                    ;*  : :    ELSE-CONTINUE
1945                    ;*  : ENDIF
1946                    ;*  : GET THE TIE BUS TABLE ADDRESS
1947                    ;*  : SINGLE STEP THE READ PATH
1948                    ;*  : INIT THE LOOP COUNT TO 0
1949                    ;*  : CALL SUBROUTINE TRK
1950                    ;*  : BGNDO
1951                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1, 2 OR 3
1952                    ;*  : :    : THEN-LOAD TIE BUS WITH 10(8)
1953                    ;*  : :    : ELSE-CLEAR TIE BUS POINTER
1954                    ;*  : :    ENDIF
1955                    ;*  : :    SINGLE STEP THE READ PATH
1956                    ;*  : :    DECREMENT THE LOOP COUNT
1957                    ;*  : :    DO UNTIL LOOP COUNT = 9
1958                    ;*  : ENDDO
1959                    ;*  : SET UP WATCHDOG TIMER COUNT
1960                    ;*  : BGNDO
1961                    ;*  : : SINGLE STEP THE READ PATH
1962                    ;*  : : DECREMENT THE WATCHDOG TIMER COUNT
1963                    ;*  : : DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1964                    ;*  : ENDDO
1965                    ;*  : IF WATCHDOG TIMER=0
1966                    ;*  : : THEN-ERROR-EXIT TEST
1967                    ;*  : : ELSE-CONTINUE
1968                    ;*  : ENDIF
1969                    ;*  : SINGLE STEP THE READ PATH
1970                    ;*  : INIT THE  LOOP COUNT TO 7
1971                    ;*  : BGNDO
1972                    ;*  : : SINGLE STEP THE READ PATH
1973                    ;*  : : COMPARE CORRECTED DATA WITH INPUT DATA
1974                    ;*  : : IF NOT EQUAL
1975                    ;*  : : : THEN-ECC ERROR
1976                    ;*  : : : ELSE-CONTINUE
1977                    ;*  : : ENDIF
1978                    ;*  : : COMPARE CORRECTED PARITY WITH INPUT PARITY
1979                    ;*  : : IF NOT EQUAL
```

```
                                ;*  : : :  THEN-ECC ERROR
                                ;*  : : :  ELSE-CONTINUE
                                ;*  : : ENDIF
                                ;*  : : DECREMENT THE LOOP COUNT
                                ;*  : : DO UNTIL THE LOOP COUNT = 0
                                ;*  : ENDDO
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : IF EXPECTED NOT EQUAL
                                ;*  : : THEN-ECC ERROR
                                ;*  : : ELSE-CONTINUE
                                ;*  : ENDIF
                                ;*  : INPUT THE ECC STATUS
                                ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
                                ;*  : : THEN-CONTINUE
                                ;*  : : ELSE-ERROR
                                ;*  : ENDIF
                                ;*  : INCREMENT THE INPUT DATA
                                ;*  : DO UNTIL INPUT DATA - 0
                                ;*  ENDDO
                                ;*ENDTST
4543                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*ECC4 MICRO TEST 02
                                ;*ECC4 MICRO ERROR 11
                                ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC4 MICRO TEST 02
                                ;*ECC4 MICRO ERROR 12
                                ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                                ;*SENT TO THE ECC
                                ;*BYTE/SCLK COUNT NUMBER   LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC4 MICRO TEST 02
                                ;*ECC4 MICRO ERROR 13
                                ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                ;*M8951, M8950'S
                                ;*DATA FROM THE ECC-INCORRECT
                                ;*DATA SHOULD NOT BE CHANGED
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*ECC4 MICRO TEST 02
```

```
                                        ;*ECC4 MICRO ERROR 14
                                        ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
                                        ;*PARITY SHOULD NOT BE CHANGED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 02
                                        ;*ECC4 MICRO ERROR 15
                                        ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 02
                                        ;*ECC4 MICRO ERROR 16
                                        ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*ECC CHARACTER SHOULD NOT BE CHANGED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 02
                                        ;*ECC4 MICRO ERROR 17
                                        ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4543                                    S
 (1)                                    ; ***************************************************************************
                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1) 4543  3E  02          7.0                   MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1) 4545  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                        ;&M8951, M8950'S

4548  3A  35  49         13.0           LDA     SUNIT
454B  D3  E0              10.0           OUT     MBSEL
454D  3E  10               7.0  TST02X:  MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
454F  D3  F0              10.0           OUT     CLKCTL
4551  3E  10               7.0           MVI     A,W.GCR
4553  D3  D3              10.0           OUT     WMCCTL
4555  3E  04               7.0           MVI     A,P.RPEN
4557  D3  4C              10.0           OUT     PENAB

4559  3E  60               7.0           MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
455B  D3  48              10.0           OUT     PDIAG           ;LCS, LWR
455D  AF                   4.0           XRA     A
455E  D3  44              10.0           OUT     TAMT
4560  3E  0A               7.0           MVI     A,10            ;DELAY
4562  3D                   4.0  1$:      DCR     A               ;50
```

```
2082  4563  C2  62  45      10.0          JNZ   1$                ;MICROSECONDS
2083  4566  AF              4.0          XRA   A                 ;CLEAR THE ACCUMULATOR
2084  4567  D3  D2          10.0          OUT   TRKENA            ;CLEAR ALL TRACKS FROM
2085  4569  D3  0A          10.0          OUT   RTIEB             ;CLEAR THE TIE BUS REGISTER
2086  456B  D3  D2          10.0          OUT   TRKENA
2087  456D  32  94  48      13.0          STA   BAD1              ;CLEAR THE INVALID
2088  4570  32  95  48      13.0          STA   BAD2              ;TRACK TRANSLATIONS
2089  4573  32  96  48      13.0          STA   BAD3              ;POINTERS
2090  4576  32  36  49      13.0          STA   INDAT             ;CLEAR THE INPUT DATA BYTE
2091  4579                                ROUT  R05H
 (1)  4579  D3  8B          10.0                OUT   R05H              ;WRITE AC INTO R05H
 (1)  457B  7F              4.0                 MOV   A,A               ;RETRY LINK
2092
2093  457C  3E  A8           7.0  TST02C: MVI   A,R.PL01.R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2094                                                                     ;SET TIE BUS JAM AND PLO DISABLE
2095  457E  D3  09          10.0          OUT   RPCTL
2096  4580  D3  08          10.0          OUT   RFIFOL            ;CLOCK THE FIFO'S
2097  4582  3E  0D           7.0          MVI   A,RCLRT           ;ISSUE CLEAR ALL COMMAND
2098  4584  D3  0B          10.0          OUT   RCMD
2099
2100  4586  00              4.0          NOP                     ;WAIT
2101  4587  00              4.0          NOP
2102  4588  00              4.0          NOP
2103  4589  00              4.0          NOP
2104  458A  00              4.0          NOP
2105
2106  458B  3E  A9           7.0          MVI   A,R.PL01!R.STPC.R.PLOD!R.TBJN   ;STOP THE READ PATH
2107                                                                     ;SET TIE BUS JAM AND PLO DISABLE
2108  458D  D3  09          10.0          OUT   RPCTL
2109  458F  CD  97  48      18.0          CALL  CLECC             ;CLEAR THE ECC CHARACTER
2110  4592  21  3F  49      10.0          LXI   H,BUFFER          ;SET UP THE BUFFER POINTER
2111  4595  16  07           7.0          MVI   D,@7              ;SET UP THE LOOP COUNT
2112  4597  3A  36  49      13.0  TST02F: LDA   INDAT             ;GET THE INPUT DATA
2113  459A  77               7.0          MOV   M,A               ;SAVE CHARACTER IN THE BUFFER
2114  459B  3C               4.0          INR   A                 ;UPDATE THE DATA
2115  459C  32  36  49      13.0          STA   INDAT             ;SAVE IT
2116  459F  23               6.0          INX   H                 ;UPDATE THE BUFFER POINTER
2117  45A0  3D               4.0          DCR   A                 ;RECREATE THE DATA FOR THE ECC SUBROUTINE
2118  45A1  CD  9C  48      18.0          CALL  ECC               ;CALCULATE ECC FOR
2119  45A4  15               4.0          DCR   D                 ;THE 7 INPUT BYTES
2120  45A5  C2  97  45      10.0          JNZ   TST02F
2121
2122  45A8  0E  03           7.0          MVI   C,@3              ;INIT THE LOOP COUNT
2123  45AA  3A  36  49      13.0  TST02I: LDA   INDAT             ;GET THE INPUT DATA
2124  45AD  06  25           7.0          MVI   B,@45             ;CHECK FOR SPECIAL CONDITION
2125  45AF  FE  03           7.0          CPI   @3                ;
2126  45B1  CA  BD  45      10.0          JZ    TST02Z            ;
2127  45B4  06  00           7.0          MVI   B,0               ;INITIALIZE THE QUOTION
2128  45B6  04               4.0  TST02W: INR   B                 ;
2129  45B7  D6  07           7.0          SUI   @7                ;DIVIDE BY 7
2130  45B9  A7               4.0          ANA   A                 ;SET THE CONDITION BITS
2131  45BA  C2  B6  45      10.0          JNZ   TST02W            ;
2132  45BD  78               4.0  TST02Z: MOV   A,B
2133  45BE                                ROUT  R05L              ;SET THE COUNTER IN THE CAS
```

```
 (1)  45BE  D3  8A        10.0                OUT     R05L          ;WRITE AC INTO R05L
 (1)  45C0  7F            4.0                 MOV     A,A           ;RETRY LINK
2134  45C1  3A  3F  49     13.0        LDA     BUFFER
2135  45C4  32  84  48     13.0        STA     TRNIN             ;FILL THE TRANSLATOR
2136  45C7  3A  40  49     13.0        LDA     BUFFER+1
2137  45CA  32  85  48     13.0        STA     TRNIN+1           ;SUBROUTINE INPUT
2138  45CD  3A  41  49     13.0        LDA     BUFFER+2
2139  45D0  32  86  48     13.0        STA     TRNIN+2           ;BUFFER
2140  45D3  3A  42  49     13.0        LDA     BUFFER+3
2141  45D6  32  87  48     13.0        STA     TRNIN+3           ;
2142  45D9  CD  B1  47     18.0        CALL    T4X5              ;TRANSLATE THE SUBGROUP
2143
2144  45DC  21  88  48     10.0        LXI     H,TRNOUT          ;GET POINTER TO TRANSLATED DATA TABLE
2145  45DF  06  05          7.0        MVI     B,@5              ;SET UP LOOP COUNT
2146  45E1  7E              7.0  TST02G: MOV     A,M               ;GET A DATA BYTE
2147  45E2  D3  40         10.0        OUT     TCMD              ;STORE DATA IN COMMAND ADDRESS
2148  45E4  23              6.0        INX     H                 ;POINT TO DATA PARITY
2149  45E5  7E              7.0        MOV     A,M               ;GET THE DATA PARITY
2150
2151  45E6  07              4.0        RLC                       ;POSITION FOR OUTPUT
2152  45E7  F6  60          7.0        ORI     P.LWR.P.LCS       ;OR IN CONTROL BITS
2153  45E9  D3  48         10.0        OUT     PDIAG             ;OUTPUT THE DATA PARITY
2154
2155  45EB  D3  08         10.0        OUT     RFIFOL            ;CLOCK DATA INTO THE FIFO'S
2156
2157  45ED  23              6.0        INX     H                 ;UPDATE THE TABLE POINTER
2158  45EE  05              4.0        DCR     B                 ;DECREMENT LOOP COUNT
2159  45EF  C2  E1  45     10.0        JNZ     TST02G            ;DO UNTIL LOOP COUNT - 0
2160
2161  45F2  3A  43  49     13.0        LDA     BUFFER+4
2162  45F5  32  84  48     13.0        STA     TRNIN             ;FILL THE FIRST THREE
2163  45F8  3A  44  49     13.0        LDA     BUFFER+5
2164  45FB  32  85  48     13.0        STA     TRNIN+1           ;BYTES OF THE TRANSLATOR
2165  45FE  3A  45  49     13.0        LDA     BUFFER+6
2166  4601  32  86  48     13.0        STA     TRNIN+2           ;SUBROUTINE WITH THE INPUT DATA
2167  4604  3A  C5  48     13.0        LDA     ECCCHR            ;STORE THE CALCULATED
2168  4607  32  87  48     13.0        STA     TRNIN+3           ;ECC CHARACTER AS THE LAST CHARACTER
2169
2170  460A  CD  B1  47     18.0        CALL    T4X5              ;TRANSLATE THE SECOND SUBGROUP
2171
2172  460D  21  88  48     10.0        LXI     H,TRNOUT          ,GET THE POINTER TO INPUT DATA TABLE
2173  4610  06  05          7.0        MVI     B,@5              ;SET UP THE LOOP COUNT
2174  4612  7E              7.0  TST02H: MOV     A,M               ;GET A DATA BYTE
2175  4613  D3  40         10.0        OUT     TCMD              ;STORE DATA IN CMD ADDRESS
2176  4615  23              6.0        INX     H                 ;POINT TO DATA PARITY BIT
2177  4616  7E              7.0        MOV     A,M               ;GET THE DATA PARITY
2178  4617  07              4.0        RLC                       ;POSITION FOR OUTPUT
2179  4618  F6  60          7.0        ORI     P.LWR.P.LCS       ;OR IN CONTROL BITS
2180  461A  D3  48         10.0        OUT     PDIAG             ;OUTPUT THE DATA PARITY
2181  461C  D3  08         10.0        OUT     RFIFOL            ;CLOCK DATA INTO THE FIFOS
2182  461E  23              6.0        INX     H                 ;UPDATE THE TABLE POINTER
2183  461F  05              4.0        DCR     B                 ;DECREMENT THE LOOP COUNT
2184  4620  C2  12  46     10.0        JNZ     TST02H            ;DO UNTIL LOOP COUNT-0
2185
```

```
2186  4623  0D                    4.0          DCR     C                ;PUT THE GROUP IN
2187  4624  C2  AA  45            10.0         JNZ     TST02I           ;THE FIFO TWICE
2188
2189  4627  3E  0B                7.0          MVI     A,DIARD          ;LOAD THE DIAGNOSTIC READ
2190  4629  D3  0B                10.0         OUT     RCMD             ;COMMAND
2191  462B  11  01  00            10.0         LXI     D,1              ;SET WATCH DOG INCREMENT
2192  462E  21  A8  FD            10.0         LXI     H,-600           ;SET WATCH DOG COUNT TO 600
2193  4631  D3  0C                10.0  2$:    OUT     RINST            ;STEP THE READ PATH
2194  4633  DB  01                10.0         IN      RPCHI            ;DATA READY SET?
2195  4635  E6  10                7.0          ANI     R.DRDY
2196  4637  C2  4C  46            10.0         JNZ     TST02A           ;YES-GO PROCESS
2197  463A  19                    10.0         DAD     D                ;WATCH DOG TIMEOUT?
2198  463B  D2  31  46            10.0         JNC     2$               ;NO-CONTINUE
2199  463E                                     ERR     RETRY,DUMMU,1
  (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  463E  CD  09  28            18.0                    CALL    ERLP             ;PROCESS ERROR - DO 2.3
  (1)        0009                                          MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4641  09                                            .BYTE   MSGN             ;MESSAGE NUMBER ID
  (1)  4642  01                                            .BYTE   1
  (1)  4643  CD  15  28            18.0           DUMMU:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4646  DA  77  47            10.0                   JC      RETRY            ;LOOP ADDRESS IF LOOP SPECIFIED
2200                                       ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2201                                       ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2202                                       ;<FATAL ERROR - MICRO TEST ABORTED
2203  4649  C3  74  47            10.0         JMP     EXIT
2204
2205  464C  CD  D8  48            18.0  TST02A: CALL    TRK              ;SET UP HARD POINTER TRACK NUMBERS
2206  464F  D3  0C                10.0         OUT     RINST
2207  4651  06  00                7.0          MVI     B,$0             ;INIT THE BIT TIME COUNT (LOOP COUNT)
2208  4653  00                    4.0  TST02B: NOP
2209  4654  3A  32  49            13.0         LDA     TIET1            ;GET FIRST TIE BUS JAM TIME
2210  4657  B8                    4.0          CMP     B                ;IS IT TIME TO JAM THE TIE BUS?
2211  4658  C2  62  46            10.0         JNZ     4$               ;NO - CONTINUE
2212  465B  3E  08                7.0          MVI     A,$8             ;LOAD ILLEGAL
2213  465D  D3  0A                10.0         OUT     RTIEB            ;5X4 POINTER
2214  465F  C3  81  46            10.0         JMP     5$               ;
2215  4662  3A  33  49            13.0  4$:    LDA     TIET2            ;GET SECOND TIE BUS JAM TIME
2216  4665  B8                    4.0          CMP     B                ;IS IT TIME TO JAM THE TIE BUS?
2217  4666  C2  70  46            10.0         JNZ     2$               ;NO-CONTINUE
2218  4669  3E  08                7.0          MVI     A,$8             ;LOAD ILLEGAL
2219  466B  D3  0A                10.0         OUT     RTIEB            ;5X4 POINTER
2220  466D  C3  81  46            10.0         JMP     5$               ;
2221  4670  3A  34  49            13.0  2$:    LDA     TIET3            ;GET THIRD TIE BUS JAM TIME
2222  4673  B8                    4.0          CMP     B                ;IS IT TIME TO JAM THE BUS?
2223  4674  C2  7E  46            10.0         JNZ     6$               ;NO-CONTINUE
2224  4677  3E  08                7.0          MVI     A,$8             ;LOAD ILLEGAL
2225  4679  D3  0A                10.0         OUT     RTIEB            ;5X4 POINTER
2226  467B  C3  81  46            10.0         JMP     5$               ;CONTINUE
2227  467E  AF                    4.0  6$:    XRA     A                ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
2228  467F  D3  0A                10.0         OUT     RTIEB            ;LOAD THE POINTER
2229  4681  D3  0C                10.0  5$:    OUT     RINST            ;STEP CLOCK
2230  4683  04                    4.0          INR     B                ;DECREMENT THE LOOP COUNT
2231  4684  78                    4.0          MOV     A,B
2232  4685  FE  09                7.0          CPI     $9               ;DONE
```

```
2233  4687  C2  53  46   10.0          JNZ   TST02B          ;DO UNTIL LOOP COUNT = ZERO
2234  468A  AF              4.0          XRA   A
2235  468B  D3  0A          10.0          OUT   RTIEB           ;CLEAR THE TIE BUS REGISTER
2236  468D  D3  0C          10.0          OUT   RINST           ;CLOCK THE READ PATH
2237  468F  D3  0C          10.0          OUT   RINST           ;CLOCK THE READ PATH
2238  4691  11  01  00      10.0          LXI   D,1             ;SET WATCHDOG TIMER INCREMENT
2239  4694  21  A8  FD      10.0          LXI   H,-600          ;SET WATCHDOG COUNT TO 600
2240  4697  D3  0C          10.0  3$:     OUT   RINST           ;SINGLE STEP THE READ PATH
2241  4699  DB  01          10.0          IN    RPCHI           ;DATA READY SET?
2242  469B  E6  10           7.0          ANI   R.DRDY
2243  469D  C2  B2  46      10.0          JNZ   TST02K          ;YES-GO PROCESS
2244  46A0  19              10.0          DAD   D               ;NO-WATCHDOG TIMEOUT?
2245  46A1  D2  97  46      10.0          JNC   3$              ;NO-CONTINUE
2246  46A4                                ERR   RETRY,DUMMV,1
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46A4  CD  09  28      18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000A                           MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A7  0A                           .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  46A8  01                           .BYTE 1
 (1)  46A9  CD  15  28      18.0  DUMMV:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46AC  DA  77  47      10.0          JC    RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2247                                ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2248                                ;>SENT TO THE ECC
2249                                ;<FATAL ERROR - MICRO TEST ABORTED
2250  46AF  C3  74  47      10.0          JMP   EXIT
2251
2252  46B2  16  01           7.0  TST02K: MVI   D,@1            ;INIT THE LOOP COUNT
2253  46B4  21  3F  49      10.0          LXI   H,BUFFER
2254  46B7  00               4.0  TST02Y: NOP
2255  46B8  7E               7.0          MOV   A,M             ;GET THE INPUT DATA
2256  46B9                                ROUT  EDATA
 (1)  46B9  D3  95          10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  46BB  7F              4.0                 MOV   A,A             ;RETRY LINK
2257  46BC  A7               4.0          ANA   A               ;SET CONDITION BITS
2258  46BD  3E  00           7.0          MVI   A,0             ;CLEAR THE ACC.
2259  46BF  E2  C3  46      10.0          JPO   1$              ;ODD # OF ONES - CONTINUE
2260  46C2  3C               4.0          INR   A               ;EVEN # OF ONES - SET PARITY
2261  46C3  32  37  49      13.0  1$:     STA   INDATP          ;STORE THE PARITY BIT
2262  46C6  D3  0C          10.0  TST02L: OUT   RINST           ;SINGLE STEP THE READ LOGIC
2263  46C8  7E               7.0          MOV   A,M
2264  46C9  23               6.0          INX   H
2265  46CA  47               4.0          MOV   B,A
2266  46CB  DB  19          10.0          IN    ECCCOR
2267  46CD  B8               4.0          CMP   B
2268  46CE  CA  D9  46      10.0          JZ    TST02M
2269  46D1                                ROUT  ADATA
 (1)  46D1  D3  94          10.0                OUT   ADATA           ;WRITE AC INIO ADATA
 (1)  46D3  7F              4.0                 MOV   A,A             ;RETRY LINK
```

```
2271  46D4                                                ERRB    RETRY,TST02M,@1
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46D4  CD  12  28        18.0                              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000B                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46D7  0B                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46D8  01                                                .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  46D9  CD  15  28        18.0                      TST02M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46DC  DA  77  47        10.0                              JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2272                                            ;>DATA FROM THE ECC-INCORRECT
2273                                            ;>DATA SHOULD NOT BE CHANGED
2274  46DF  DB  15               10.0                 IN      RPSTA           ;GET THE ECC PARITY BIT
2275  46E1  E6  80                7.0                 ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2276  46E3  07                    4.0                 RLC                     ;POSITION BIT FOR COMPARE
2277  46E4  4F                    4.0                 MOV     C,A             ;SAVE IN REG. C
2278  46E5  3A  37  49            13.0                LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2279  46E8  A7                    4.0                 ANA     A               ;SET THE CONDITION BITS
2280  46E9  CA  00  47            10.0                JZ      TST02QO         ;GO CHECK FOR PARITY=0
2281  46EC  79                    4.0                 MOV     A,C             ;ELSE CHECK FOR PARITY=1
2282  46ED  FE  01                7.0                 CPI     @1
2283  46EF  CA  F7  46            10.0                JZ      TST02RG
2284  46F2                                            ERR     RETRY,TST02RG,@1
 (1)                                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46F2  CD  09  28        18.0                              CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46F5  0C                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46F6  01                                                .BYTE   @1
 (1)  46F7  CD  15  28        18.0                      TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46FA  DA  77  47        10.0                              JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2285                                            ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2286                                            ;>PARITY SHOULD NOT BE CHANGED
2287                                            ;<
2288  46FD  C5  14  47            10.0                JMP     TST02SZ         ;CONTINUE WITH TEST
2289  4700  79                    4.0         TST02QO: MOV    A,C             ;CHECK FOR PARITY = ZERO
2290  4701  A7                    4.0                 ANA     A               ;SET CONDITION BITS
2291  4702  CA  0E  47            10.0                JZ      TST02TP         ;CONTINUE IF ZERO
2292  4705  78                    4.0                 MOV     A,B
2293  4706                                            ROUT    R05L
 (1)  4706  D3  8A                10.0                        OUT     R05L            ;WRITE AC INTO R05L
 (1)  4708  7F                    4.0                 MOV     A,A             ;RETRY LINK
2294  4709                                            ERR     RETRY,TST02TP,@1
 (1)                                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4709  CD  09  28        18.0                              CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000D                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  470C  0D                                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  470D  01                                                .BYTE   @1
 (1)  470E  CD  15  28        18.0                      TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4711  DA  77  47        10.0                              JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2295                                            ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2296                                            ;<
2297  4714  7A                    4.0         TST02SZ: MOV    A,D             ;GET THE CHARACTER COUNT
2298  4715  3C                    4.0                 INR     A               ;INCREMENT
2299  4716  FE  08                7.0                 CPI     $8              ;DONE?
2300  4718  CA  1F  47            10.0                JZ      TST02N          ;YES-GO CHECK ECC
2301  471B  57                    4.0                 MOV     D,A             ;NO-CONTINUE
```

```
2302  471C   C3   B7   46      10.0          JMP      TST02Y
2303
2304  471F   D3   0C           10.0  TST02N: OUT      RINST              ;SINGLE STEP THE READ PATH
2305  4721   3A   C5   48      13.0          LDA      ECCCHR             ;GET THE EXPECTED ECC CHARACTER
2306  4724                                   ROUT     EDATA              ;SAVE IN CAS
 (1)  4724   D3   95           10.0                   OUT      EDATA              ;WRITE AC INTO EDATA
 (1)  4726   7F                 4.0                   MOV      A,A                ;RETRY LINK
2307  4727   47                 4.0          MOV      B,A                ;SAVE IN REGISTER B
2308  4728   DB   19           10.0          IN       ECCCOR
2309  472A                                   ROUT     ADATA
 (1)  472A   D3   94           10.0                   OUT      ADATA              ;WRITE AC INTO ADATA
 (1)  472C   7F                 4.0                   MOV      A,A                ;RETRY LINK
2310  472D   B8                 4.0          CMP      B
2311  472E   CA   36   47      10.0          JZ       TST02U
2312  4731                                   ERR      RETRY,TST02U,1
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4731   CD   09   28      18.0                            CALL     ERLP               ;PROCESS ERROR - DO 2.3
 (1)        000E                                               MSGN     =        MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4734   0E                                                .BYTE    MSGN               ;MESSAGE NUMBER ID
 (1)  4735   01                                                .BYTE    1
 (1)  4736   CD   15   28      18.0                   TST02U::          CALL     CKLOP             ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4739   DA   77   47      10.0                            JC       RETRY              ;LOOP ADDRESS IF LOOP SPECIFIED
2313                                 ;>CORRECTED ECC CHARACTER INCORRECT
2314                                 ;>ECC CHARACTER SHOULD NOT BE CHANGED
2315  473C   DB   1A           10.0          IN       ECCSTA             ;GET THE ECC STATUS
2316  473E   E6   0F            7.0          ANI      @17                ;REMOVE UNWANTED BITS
2317  4740   FE   04            7.0          CPI      E.UNC              ;IS UNCORRECTABLE BIT SET?
2318  4742   CA   52   47      10.0          JZ       TST02V
2319  4745                                   ROUT     ADATA
 (1)  4745   D3   94           10.0                   OUT      ADATA              ;WRITE AC INTO ADATA
 (1)  4747   7F                 4.0                   MOV      A,A                ;RETRY LINK
2320  4748   3E   04            7.0          MVI      A,E.UNC
2321  474A                                   ROUT     EDATA
 (1)  474A   D3   95           10.0                   OUT      EDATA              ;WRITE AC INTO EDATA
 (1)  474C   7F                 4.0                   MOV      A,A                ;RETRY LINK
2322  474D                                   ERRB     RETRY,TST02V,1
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  474D   CD   12   28      18.0                            CALL     ERLPB              ;PROCESS ERROR - DO 2.3
 (1)        000F                                               MSGN     =        MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4750   0F                                                .BYTE    MSGN               ;MESSAGE NUMBER ID
 (1)  4751   01                                                .BYTE    1                  ;PRINT ROUTINE NUMBER
 (1)  4752   CD   15   28      18.0                   TST02V::          CALL     CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4755   DA   77   47      10.0                            JC       RETRY              ;LOOP ADDRESS IF LOOP SPECIFIED
2323                                 ;>ECC STATUS INCORRECT AFTER A DATA GROUP
2324  4758   3A   36   49      13.0          LDA      INDAT
2325  475B   FE   03            7.0          CPI      @3                 ;DONE
2326  475D   C2   7C   45      10.0          JNZ      TST02C
```

```
                                                ENDTST  TST02X
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
4760                                                            CALL    REQST
4760    CD  06  28      18.0                                    .BYTE                   ;DATA PATTERN NUMBER
4763    00                                                      .WORD                   ;SYSTEM '*' COUNT
4764    00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
4766    00  00                                                  .BYTE                   ;DATA COMPARE FLAG IF =1
4768    00                                                      .BYTE   7               ;REQUEST CODE
4769    07
476A    3A  9A  4F      13.0            LDA     ITERA           ;GET ITERATION COUNT
476D    3D               4.0            DCR     A               ;DOWNCOUNT
476E    32  9A  4F      13.0            STA     ITERA           ;SAVE COUNT
4771    F2  4D  45      10.0            JP      TST02X          ;DO TEST UNTIL TILL = 0

4774    C3  18  28      10.0    EXIT:   JMP     TSTEND
4777    3A  36  49      13.0    RETRY:  LDA     INDAT
477A    D6  07           7.0            SUI     @7
477C    32  36  49      13.0            STA     INDAT
477F    C3  7C  45      10.0            JMP     TST02C
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
2337  4782                              SSUB
                                        ;*********************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
2339  4782                              SD
                                        ;*********************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
2343  4782                              SP
                                        ;*********************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
2352  4782                              S
                                        ;*********************************************************************************
2353  4782  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
2354  4783  C5              12.0        PUSH    B
2355  4784  06  00           7.0        MVI     B,0             ;START TO CLEAR AT PORT #0
2356  4786  DB  E0          10.0  CLRLP: IN      INTSTA          ;GET MB SELECT INFO
2357  4788  E6  80           7.0        ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
2358  478A  B0               4.0        ORA     B               ;ADD IN THE SELECTED PORT #
2359  478B  D3  E0          10.0        OUT     MBSEL           ;RESET TO THIS PORT
2360  478D  3E  80           7.0        MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
2361  478F  D3  40          10.0        OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
2362  4791  AF               4.0        XRA     A               ;CLEAR TU COMMAND A
2363  4792  D3  40          10.0        OUT     TCMD            ;
2364  4794  3E  81           7.0        MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
2365  4796  D3  40          10.0        OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
2366  4798  3E  00           7.0        MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
2367  479A  D3  40          10.0        OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
2368  479C  AF               4.0        XRA     A
2369  479D  D3  44          10.0        OUT     TAMT            ;CLEAR AMTIE WORD
2370  479F  D3  48          10.0        OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
2371  47A1  D3  4C          10.0        OUT     PENAB           ;CLEAR PORT ENABLE WORD
2372  47A3  04               4.0        INR     B               ;POINT TO THE NEXT PORT TO CLEAR
2373  47A4  78               4.0        MOV     A,B
2374  47A5  FE  04           7.0        CPI     4               ;DONE?
2375  47A7  C2  86  47       10.0        JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
2376  47AA  C1               10.0        POP     B               ;RESET B & C
2377  47AB  F1               10.0        POP     PSW             ;ALL DONE
2378  47AC  C9               10.0        RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
47AD                            SSUB
                                ;*********************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
47AD                            SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE, AT THE ENTRY
                                ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                ;*BUT NOT CORRECT FOR THE INPUT DATA SIMULATING BAD DATA.
47AD                            SP
                                ;*********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    CLEAR THE BAD TRANSLATION POINTERS
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 1
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    INCREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 10
                                ;*    ENDDO
                                ;*ENDSUB
47AD                            S
                                ;*********************************************************************************

                                
47AD  F5            12.0  BADT4X5: PUSH   PSW             ;SAVE REGISTER A + PSW
47AE  C3  BC  47    10.0           JMP    BAD
47B1  F5            12.0  T4X5:    PUSH   PSW             ;SAVE REGISTER A + PSW
47B2  AF             4.0           XRA    A               ;CLEAR A
47B3  32  94  48    13.0           STA    BAD1            ;CLEAR BAD TRANSLATION 1
47B6  32  95  48    13.0           STA    BAD2            ;CLEAR BAD TRANSLATION 2
47B9  32  96  48    13.0           STA    BAD3            ;CLEAR BAD TRANSLATION 3
47BC  C5            12.0  BAD:     PUSH   B               ;SAVE REGISTER B + C
47BD  D5            12.0           PUSH   D               ;SAVE REGISTER D + E
47BE  E5            12.0           PUSH   H               ;SAVE REGISTER H + L
```

```
2424                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
2425                                        ;CLEAR THE TRANSLATED DATA TABLE.
2426  47BF  06  0A          7.0            MVI   B,10          ;SET UP LOOP COUNT
2427  47C1  11  88  48      10.0           LXI   D,TRNOUT      ;GET POINTER TO TRANSLATED DATA TABLE
2428  47C4  AF               4.0            XRA   A           ;CLEAR A
2429  47C5  12               7.0  D4X5:     STAX  D           ;STORE IN THE TABLE
2430  47C6  13               6.0            INX   D           ;UPDATE TABLE POINTER
2431  47C7  05               4.0            DCR   B           ;DECREMENT LOOP COUNT
2432  47C8  C2  C5  47       10.0           JNZ   D4X5        ;DO UNTIL LOOP COUNT=0
2433  47CB  0E  01           7.0            MVI   C,1         ;SET UP TRACK COUNT
2434  47CD  06  04           7.0  B4X5:     MVI   B,4         ;SET UP BIT COUNT
2435  47CF  11  84  48       10.0           LXI   D,TRNIN
2436  47D2  AF               4.0            XRA   A           ;CLEAR THE GROUP POSITION COUNT
2437  47D3  32  92  48       13.0           STA   GP4X5
2438  47D6  1A               7.0  C4X5:     LDAX  D           ;GET A DATA BYTE
2439  47D7  A7               4.0            ANA   A           ;SET CONDITION BITS
2440  47D8  E2  DC  47       10.0           JPO   PO4X5       ;ODD PARITY LEAVE CARRY CLEAR
2441  47DB  37               4.0            STC               ;EVEN PARITY SET CARRY
2442  47DC  1F               4.0  PO4X5:     RAR               ;SHIFT OUT DESIRED BIT
2443  47DD  12               7.0            STAX  D           ;STORE RESULT BACK IN TEMP TABLE
2444  47DE  3A  92  48       13.0           LDA   GP4X5       ;GET THE GROUP POSITION BYTE
2445  47E1  17               4.0            RAL               ;PUT IN THIS DATA BIT
2446  47E2  32  92  48       13.0           STA   GP4X5       ;SAVE THE UPDATED GROUP POSITION BYTE
2447  47E5  13               6.0            INX   D           ;UPDATE THE TABLE POINTER
2448  47E6  05               4.0            DCR   B           ;DECREMENT THE BIT COUNT
2449  47E7  C2  D6  47       10.0           JNZ   C4X5        ;DO UNTIL ALL 4 BITS ARE RECEIVED
2450
2451                                        ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2452
2453                                        ;PERFORM A DOUBLE PRECISION SHIFT RIGHT ON THE TRANSLATED DATA
2454  47EA  06  05           7.0            MVI   B,5         ;SET UP LOOP COUNT
2455  47EC  11  88  48       10.0           LXI   D,TRNOUT    ;GET POINTER TO TRANSLATED DATA TABLE
2456  47EF  13               6.0  E4X5:     INX   D           ;POINT TO PARITY BIT
2457  47F0  1A               7.0            LDAX  D           ;GET PARITY BIT
2458  47F1  A7               4.0            ANA   A           ;CLEAR THE CARRY BIT
2459  47F2  1F               4.0            RAR               ;SHIFT TO THE CARRY BIT
2460  47F3  12               7.0            STAX  D           ;STORE IT BACK
2461  47F4  1B               6.0            DCX   D           ;DECREMENT TO DATA BITS
2462  47F5  1A               7.0            LDAX  D           ;GET THE DATA BITS
2463  47F6  1F               4.0            RAR               ;SHIFT IN CARRY BITS
2464  47F7  12               7.0            STAX  D           ;STORE BACK IN TABLE
2465  47F8  13               6.0            INX   D           ;POINT TO NEXT TABLE ENTRY
2466  47F9  13               6.0            INX   D           ;
2467  47FA  05               4.0            DCR   B           ;DECREMENT LOOP COUNT
2468  47FB  C2  EF  47       10.0           JNZ   E4X5        ;DO UNTIL LOOP COUNT=0
2469  47FE  3A  94  48       13.0           LDA   BAD1        ;IS A BAD POINTER SPECIFIED?
2470  4801  A7               4.0            ANA   A           ;SET CONDITION BITS
2471  4802  CA  09  48       10.0           JZ    BAD1C       ;NO-CONTINUE
2472  4805  B9               4.0            CMP   C           ;YES-IS IT THIS DATA POSITION?
2473  4806  CA  1F  48       10.0           JZ    INV4X5      ;YES-GO PERFORM INCORRECT TRANSLATION
2474  4809  3A  95  48       13.0 BAD1C:    LDA   BAD2        ;IS A BAD POINTER SPECIFIED
2475  480C  A7               4.0            ANA   A           ;SET CONDITION BITS
2476  480D  CA  14  48       10.0           JZ    BAD2C       ;NO-CONTINUE
2477  4810  B9               4.0            CMP   C           ;YES-IS IT THIS DATA POSITION
```

```
2478  4811  CA  1F  48   10.0          JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2479  4814  3A  96  48   13.0  BAD2C:  LDA     BAD3            ;IS A BAD POINTER SPECIFIED
2480  4817  A7              4.0          ANA     A               ;SET CONDITION BITS
2481  4818  CA  26  48   10.0          JZ      BAD3C           ;NO-PERFORM PROPER TRANSLATION
2482  481B  B9              4.0          CMP     C               ;YES-IS IT THIS DATA POSITION
2483  481C  C2  26  48   10.0          JNZ     BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
2484  481F  3A  92  48   13.0  INV4X5: LDA     GP4X5           ;GET THE GROUP POSITION COUNT
2485  4822  3C              4.0          INR     A               ;INCREMENT-TO INCORRECT TRANSLATION
2486  4823  32  92  48   13.0          STA     GP4X5           ;STORE IT
2487  4826  11  74  48   10.0  BAD3C:  LXI     D,TAB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
2488  4829  26  00            7.0          MVI     H,0             ;CLEAR REGISTER H
2489  482B  3A  92  48   13.0          LDA     GP4X5           ;GET GROUP POSITION COUNT
2490  482E  E6  0F            7.0          ANI     $0F
2491  4830  6F              4.0          MOV     L,A
2492  4831  19             10.0          DAD     D               ;ADD GROUP POSITION COUNT TO TABLE
2493  4832  7E              7.0          MOV     A,M             ;GET TRANSLATED DATA
2494  4833  32  93  48   13.0          STA     TRNTMP
2495                                 ;''TRNTMP'' = THE TRANSLATED DATA
2496
2497  4836  11  88  48   10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2498  4839  06  05            7.0          MVI     B,5             ;SET UP LOOP COUNT
2499  483B  3A  93  48   13.0  F4X5:   LDA     TRNTMP          ;GET TRANSLATED DATA
2500  483E  13              6.0          INX     D               ;POINT TO PARITY POSITION
2501  483F  A7              4.0          ANA     A               ;CLEAR CARRY BIT
2502  4840  1F              4.0          RAR                     ;SHIFT OUT A BIT
2503  4841  32  93  48   13.0          STA     TRNTMP          ;STORE TRANSLATED DATA
2504  4844  1A            7.0          LDAX    D               ;GET THE PARITY BIT (ALWAYS ZERO)
2505  4845  17              4.0          RAL                     ;ROLL IN THE CARRY BIT
2506  4846  12            7.0          STAX    D               ;STORE AS THE PARITY BIT
2507  4847  13              6.0          INX     D               ;UPDATE TABLE POINTER
2508  4848  05              4.0          DCR     B               ;DECREMENT THE LOOP COUNT
2509  4849  C2  3B  48   10.0          JNZ     F4X5            ;DO UNTIL LOOP COUNT=0
2510
2511                                 ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2512
2513  484C  0C              4.0          INR     C               ;DECREMENT THE TRACK COUNT
2514  484D  79              4.0          MOV     A,C
2515  484E  FE  0A            7.0          CPI     10
2516  4850  C2  CD  47   10.0          JNZ     B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2517  4853  E1             10.0          POP     H               ;RESTORE REGISTER H + L
2518  4854  D1             10.0          POP     D               ;RESTORE REGISTER D + E
2519  4855  C1             10.0          POP     B               ;RESTORE REGISTER B + C
2520  4856  F1             10.0          POP     PSW             ;RESTORE REGISTER A + PSW
2521  4857  C9             10.0          RET                     ;RETURN TO USER
2522
```

```
                                        .SBTTL  SUBROUTINE CALCULATE POINTER
                                        ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                                        ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                        ;*
                                        ;*      INPUT/EXPECTED
                                        ;*          DATA                        POINTER GENERATED
                                        ;*      --------------                  -----------------
                                        ;*
                                        ;*         XX00XXXX                     HISTORY
                                        ;*         XX01XXXX                     PHASE TIE
                                        ;*         XX10XXXX                     AMTIE
                                        ;*         XX11XXXX                     ILLEGAL 5X4
                                        ;*
                                        ;*      X = DON'T CARE BITS
4858                                    SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------

                                        POINTER: PUSH   PSW             ;SAVE THE ACCUMULATOR
                                                 PUSH   B               ;SAVE B + C
                                                 LDA    INDAT           ;GET INPUT DATA
                                                 ANI    $30             ;REMOVE THE WORKING BITS
                                                 RLC                    ;POSITION THE DATA
                                                 RLC
                                                 RLC
                                                 RLC
                                                 LXI    H,POINTBL       ;GET BASE ADDRESS OF TABLE
                                                 MOV    C,A
                                                 MVI    B,0
                                                 DAD    B
                                                 MOV    A,M             ;GET THE ECC POINTER
                                                 OUT    RTIEB           ;SET THE TIE BUS VALUE
                                                 POP    B               ;RESTORE B + C
                                                 POP    PSW             ;RESTORE ACCUMULATOR
                                                 RET

                                        POINTBL: .BYTE  1
                                                 .BYTE  2
                                                 .BYTE  4
                                                 .BYTE  8
```

```
2564                                    .SBTTL  TABLE 4 X 5 TRANSLATION
2565
2566                                    ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2567                                    ;   INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2568                                    ;    1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
2569                                    ;
2570                                    ;         0000                              10011
2571                                    ;         0001                              11011
2572                                    ;         0010                              01001
2573                                    ;         0011                              11001
2574                                    ;         0100                              10111
2575                                    ;         0101                              10101
2576                                    ;         0110                              01101
2577                                    ;         0111                              11101
2578                                    ;         1000                              01011
2579                                    ;         1001                              10010
2580                                    ;         1010                              01010
2581                                    ;         1011                              11010
2582                                    ;         1100                              01111
2583                                    ;         1101                              10110
2584                                    ;         1110                              01110
2585                                    ;         1111                              11110
2586
2587  4874  13                          TAB4X5: .BYTE   @23
2588  4875  1B                                  .BYTE   @33
2589  4876  09                                  .BYTE   @11
2590  4877  19                                  .BYTE   @31
2591  4878  17                                  .BYTE   @27
2592  4879  15                                  .BYTE   @25
2593  487A  0D                                  .BYTE   @15
2594  487B  1D                                  .BYTE   @35
2595  487C  0B                                  .BYTE   @13
2596  487D  12                                  .BYTE   @22
2597  487E  0A                                  .BYTE   @12
2598  487F  1A                                  .BYTE   @32
2599  4880  0F                                  .BYTE   @17
2600  4881  16                                  .BYTE   @26
2601  4882  0E                                  .BYTE   @16
2602  4883  1E                                  .BYTE   @36
2603
```

```
2605                                    .SBTTL  SUBROUTINE VARIABLES
2606
2607  4884 0004                         TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2608                                                                    ;BE TRANSLATED
2609  4888 000A                         TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2610                                                                    ;TRANSLATION
2611  4892   00                         GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2612                                                                    ;BE TRANSLATED
2613  4893   00                         TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2614  4894   00                         BAD1:   .BYTE   0               ;BAD TRANSLATION POINTER 1
2615  4895   00                         BAD2:   .BYTE   0               ;BAD TRANSLATION POINTER 2
2616  4896   00                         BAD3:   .BYTE   0               ;BAD TRANSLATION POINTER 3
2617
```

```
                                        .SBTTL  SUBROUTINE CLEAR ECC
                                        S
                                        ; ***************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
                                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
                                        S
                                        ; ***************************************************************************

4897  AF                      4.0 CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
4898  32  C5  48              13.0        STA     ECCCHR          ;CLEAR THE ECC CHARACTER
489B  C9                      10.0        RET                     ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        S
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        S
                                        ; ************************************************************************
                                        
                                12.0    ECC:    PUSH    H               ;SAVE H&L
                                12.0            PUSH    D               ;SAVE D&E
                                12.0            PUSH    PSW             ;SAVE THE ACCUMULATOR
                                10.0            LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
                                 7.0            XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
                                 4.0            MOV     E,A             ;SAVE XOR RESULT IN E
                                 7.0            ANI     $10             ;IS BIT #4 OF RESULT SET
                                 4.0            MOV     A,E             ;RESTORE XOR RESULT FROM B
                                10.0            JZ      ECC1            ;CONTINUE IF BIT #4 RESET
                                 7.0            XRI     $23             ;ELSE-XOR WITH 23
                                 4.0    ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                                                        ;
                                 4.0            XRA     A               ;CLEAR A
                                 4.0            MOV     C,A             ;CLEAR THE TRANSLATE RESULT
                                10.0            LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
                                18.0            CALL    TRANS           ;TRANSLATE THE BITS
                                 4.0            MOV     A,C             ;GET THE TRANSLATED RESULT
                                13.0            STA     ECCCHR          ;STORE RESULT
                                10.0            POP     PSW             ;RESTORE THE ACCUMULATOR
                                10.0            POP     D               ;RESTORE D&E
                                10.0            POP     H               ;RESTORE H&L
                                10.0            RET
                                
                                        ECCTBL: $08                     ;BIT 0 = POSITION 3
                                                $20                     ;BIT 1 = POSITION 5
                                                $02                     ;BIT 2 = POSITION 1
                                                $40                     ;BIT 3 = POSITION 6
                                                $80                     ;BIT 4 = POSITION 7
                                                $01                     ;BIT 5 = POSITION 0
                                                $10                     ;BIT 6 = POSITION 4
                                                $04                     ;BIT 7 = POSITION 2
                                
                                        ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;
48C6  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
48C8  7B              4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
48C9  A0              4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
48CA  CA  D0  48     10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
48CD  79              4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
48CE  B6              7.0          ORA   M            ;'OR' IN NEW POSITION
48CF  4F              4.0          MOV   C,A          ;SAVE RESULT

48D0  78              4.0  TRANS2: MOV   A,B
48D1  07              4.0          RLC                ;POSITION MASK TO NEXT BIT
48D2  47              4.0          MOV   B,A
48D3  D8             12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
48D4  23              6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
48D5  C3  C8  48     10.0          JMP   TRANS1       ;PROCESS NEXT BIT
```

```
                                .SBTTL  SUBROUTINE TRK
48D8                            SSUB
                                ;*********************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*GENERATE BAD TRACK TRANSLATION
48D8                            SD
                                ;*********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE GENRATES THREE TRACK NUMBERS FOR THE BADT4X5 SUBROUTINE
                                ;*BASED ON THE VALUE OF INDAT AND THREE TIE BUS TIMES
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND BITS SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITIONS
                                ;*          DATA                WITH ECC CORRECTION
                                ;*      --------------          -------------------
                                ;*
                                ;*         XXXX0000                    0,1,2
                                ;*         XXXX0001                    1,2,3
                                ;*         XXXX0010                    2,3,4
                                ;*         XXXX0011                    3,4,5
                                ;*         XXXX0100                    4,5,6
                                ;*         XXXX0101                    5,6,7
                                ;*         XXXX0110                    6,7,P
                                ;*         XXXX0111                    7,P,0
                                ;*         XXXX1000                    P,0,1
                                ;*         XXXX1001                    1,2,3
                                ;*         XXXX1010                    2,3,4
                                ;*         XXXX1011                    3,4,5
                                ;*         XXXX1100                    4,5,6
                                ;*         XXXX1101                    5,6,7
                                ;*         XXXX1110                    6,7,P
                                ;*         XXXX1111                    7,P,0
                                ;*
                                ;*      X = DON'T CARE BITS
48D8                            S
                                ;*********************************************************************************

48D8  3A 36 49      13.0  TRK:  LDA   INDAT         ;GET INPUT DATA
48DB  E6 0F          7.0        ANI   $0F           ;REMOVE 4 BITS
48DD  FE 08          7.0        CPI   $8            ;PARITY TRACK?
48DF  CA E4 48      10.0        JZ    TRKA          ;YES
48E2  E6 07          7.0        ANI   $7            ;NO-IGNORE BIT3
48E4  3C             4.0  TRKA: INR   A             ;INCREMENT FOR SUBROUTINE
48E5  32 94 48      13.0        STA   BAD1          ;SAVE IT.
48E8  3D             4.0        DCR   A             ;PREPARE TO USE AS TABLE INDEX
48E9  21 29 49      10.0        LXI   H,TIMTBL      ;INCORRECT TIE BUS TIME TABLE START
48EC  5F             4.0        MOV   E,A           ;LOW HALF OF TABLE INDEX
48ED  16 00          7.0        MVI   D,0           ;ZERO HIGH HALF
48EF  19            10.0        DAD   D             ;PRODUCE TABLE ENTRY ADDRESS
```

```
2743  48F0  7E          7.0          MOV   A,M          ;GET INCORRECT TIE BUS TIME FROM TABLE
2744  48F1  32  32  49  13.0          STA   TIFT1        ;SAVE FIRST TIE BUS JAM TIME
2745  48F4  3A  94  48  13.0          LDA   BAD1
2746  48F7  3C          4.0          INR   A            ;GENERATE VALUE FOR BAD2
2747  48F8  FE  0A      7.0          CPI   $A           ;TOO BIG?
2748  48FA  C2  FF  48  10.0          JNZ   TRKB         ;JUMP IF OK
2749  48FD  3E  01      7.0          MVI   A,@1         ;YES
2750  48FF  32  95  48  13.0  TRKB:   STA   BAD2         ;SAVE IT
2751  4902  3D          4.0          DCR   A            ;PREPARETO USE AS TABLF INDEX
2752  4903  21  29  49  10.0          LXI   H,TIMTBL     ;TIE BUS TIME TABLE START
2753  4906  5F          4.0          MOV   E,A          ;LOW HALF OF TABLE INDEX
2754  4907  16  00      7.0          MVI   D,0          ;CLEAR UPPER HALF
2755  4909  19          10.0         DAD   D            ;PRODUCE TABLE ENTRY ADDRESS
2756  490A  7E          7.0          MOV   A,M          ;GET SECOND TIE BUS JAM TIME
2757  490B  32  33  49  13.0          STA   TIET2        ;SAVE IT
2758  490E  3A  95  48  13.0          LDA   BAD2
2759  4911  3C          4.0          INR   A
2760  4912  FE  0A      7.0          CPI   $A
2761  4914  C2  19  49  10.0          JNZ   TRKC
2762  4917  3E  01      7.0          MVI   A,@1
2763  4919  32  96  48  13.0  TRKC:   STA   BAD3
2764  491C  3D          4.0          DCR   A
2765  491D  21  29  49  10.0          LXI   H,TIMTBL
2766  4920  5F          4.0          MOV   E,A
2767  4921  16  00      7.0          MVI   D,0
2768  4923  19          10.0         DAD   D
2769  4924  7E          7.0          MOV   A,M
2770  4925  32  34  49  13.0          STA   TIET3
2771  4928  C9          10.0         RET
2772  4929  02                TIMTBL: .BYTE  2          ; TIE BUS TIME FOR BIT 0
2773  492A  03                        .BYTE  3          ; TIE BUS TIME FOR BIT 1
2774  492B  00                        .BYTE  0          ; TIE BUS TIME FOR BIT 2
2775  492C  01                        .BYTE  1          ; TIE BUS TIME FOR BIT 3
2776  492D  04                        .BYTE  4          ; TIE BUS TIME FOR BIT 4
2777  492E  08                        .BYTE  8          ; TIE BUS TIME FOR BIT 5
2778  492F  07                        .BYTE  7          ; TIE BUS TIME FOR BIT 6
2779  4930  05                        .BYTE  5          ; TIE BUS TIME FOR BIT 7
2780  4931  06                        .BYTE  6          ; TIE BUS TIME FOR BIT P
2781
2782
```

```
2784                                    .SBTTL  PROGRAM VARIABLES
2785
2786  4932  00                          TIET1:  .BYTE   0               ;FIRST TIE JAM TIME
2787  4933  00                          TIET2:  .BYTE   0               ;SECOND TIE BUS JAM TIME
2788  4934  00                          TIET3:  .BYTE   0               ;THIRD TIE BUS JAM TIME
2789  4935  00                          SUNIT:  .BYTE   0               ;TU PORT #
2790  4936  00                          INDAT:  .BYTE   0               ;DATA BYTE TO ECC
2791  4937  00                          INDATP: .BYTE   0               ;DATA BYTE PARITY
2792  4938  00                          UNITMP: .BYTE   0               ;UNIT MAP
2793  4939  00  00                      AMTMSK: .WORD   0               ;AMTIE MASK WORD
2794  493B  00                          DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2795  493C  00                          DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2796  493D  00                          DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2797  493E  00                          DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2798  493F  0007                        BUFFER: .BLKB   7               ;DATA BYTE STORAGE
2799        0000                              .END
```

```
A       %0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
AMTMSK   4939   ARAIDF= 0098   ASAVE   4F9B   ATTCD   4F97
AXNUM    4F91   B      =%0000   BAD     47BC   BADST  = 0090
BADT4X   47AD   BAD1     4894   BAD1C   4809   BAD2    4895
BAD2C    4814   BAD3     4896   BAD3C   4826   BIT0   = 0001
BIT1   = 0002   BIT15  = 8000   BIT2   = 0004   BIT3   = 0008
BIT4   = 0010   BIT5   = 0020   BIT6   = 0040   BIT7   = 0080
BIT8   = 0100   BIT9   = 0200   BRKPBC= 4F0A   BRKRAM= 4F10
BRKSTR= 4E60   BRKXCT= 4F00   BSAVE   4F9C   BUFFER  493F
BYTCNT= 00D4   BYTEH    4F24   BYTEL   4F23   B4X5    47CD
C       =%0001   CASCT    4F21   CASCTL= 00A0   CASSTA= 00A0
CATTH  = 0089   CATTL  = 0088   CBUSST= 00A1   CBYTH  = 008B
CBYTL  = 008A   CDG1H  = 0087   CDG1L  = 0086   CDG2H  = 0093
CDG2L  = 0092   CDG3H  = 0095   CDG3L  = 0094   CDVTH  = 008D
CDVTL  = 008C   CHPTIE= 0028   CHOTIE= 0020   CH1TIE= 0021
CH2TIE= 0022   CH3TIE= 0023   CH4TIE= 0024   CH5TIE= 0025
CH6TIE= 0026   CH7TIE= 0027   CKLOP  = 2815   CLEAR   4782
CLECC    4897   CLKCTL= 00F0   CLOCK   4F26   CLRLP   4786
CMC0H  = 0099   CMC0L  = 0098   CMC1H  = 009B   CMC1L  = 009A
CMC2H  = 009D   CMC2L  = 009C   CMC3H  = 009F   CMC3L  = 009E
CMINH  = 0097   CMINL  = 0096   CNTCTL= 00D7   CRCWRD= 0018
CSAVE    4F9D   CSRLH  = 0091   CSRLL  = 0090   CTCH   = 0085
CTCL   = 0084   CTSTH  = 008F   CTSTL  = 008E   CXCTH  = 0081
CXCTL  = 0080   CXINH  = 0083   CXINL  = 0082   C.     = 0001
C.AVAI= 0080   C.DP   = 0008   C.DSE  = 0010   C.DTU  = 0003
C.DVA  = 0008   C.FAIL= 00FC   C.FMT  = 0070   C.FNCT= 003E
C.GO   = 0001   C.INTC= 00FE   C.MAIN= 0020   C.NSA  = 0080
C.RCT  = 00FC   C.SER  = 0080   C.SHR  = 0040   C.SKPC= 000F
C.TAPE= 0040   C.WCS  = 0002   C4X5    47D6   D       =%0002
DATAA    493B   DATAAP  493C   DATACT= 00D0   DATAE   493D
DATAEP   493E   DBUS    4F28   DBUSCT= 00C0   DBUSST= 00C0
DDRA   = 00D8   DDRAIN= 0010   DDRB   = 00D9   DDRBIN= 000?
DDRC   = 00DA   DDRCIN= 0001   DDRCO  = 0088   DDRCTL= 00DB
DIAFLG   4F22   DIAGPG= 4300   DIAGRM= 4F90   DIARD  = 000B
DONE1  = 0045   DONINT= 0010   DSAVE   4F9E   DSE    = 0006
DUMMU    4643 G DUMMV    46A9 G DUMMW   4427 G DUMMX   447A G
DUMMY    431E G D.ATH0= 0001   D.ATH1= 0002   D.EOTD= 0010
D.LAGC= 0020   D.NOTW= 0040   D.NTHR= 0004   D.TACH= 0008
D.WR4  = 0080   D4X5    47C5   E       =%0003   ECC     489C
ECCBAD= 0042   ECCCHR  48C5   ECCCOR= 0019   ECCOK  = 0041
ECCSTA= 001A   ECCTBL  488D   ECCTST= 000E   ECCI    48AC
EDATA  = 0095   EOTCLR= 0003   ERFLG   4F93   ERLP   = 2809
ERLPA  = 280F   ERLPB  = 2812   ERLPE  = 280C   ERNUM   4F90
ERRCNT= 00D6   ESAVE   4F9F   EXIT    4774   E.ACRC= 0010
E.AMT  = 0020   E.CDP  = 0080   E.CRC  = 0080   E.PNTR= 0008
E.RPE  = 0040   E.STEC= 0001   E.TTEC= 0002   E.UNC  = 0004
E4X5     47EF   FIFORD= 006A   FORMAT  4F25   FOUND   4344
FWDTST= 0061   F4X5    483B   GCRID  = 0089   GCRSET= 0002
GOODTM= 0092   GP4X5   4892   H       =%0004   HLSAVE  4FA0
IE     = 0008   INDAT   4936   INDATP  4937   INTSTA= 00E0
INV4X5   481F   ITERA   4F9A   I.PWR  = 0020   I.RMPE= 0040
I5.5   = 0010   I6.5   = 0020   I7.5   = 0040   KCALL  = 005F
KCLR   = 007B   KDEP   = 003F   KENAB  = 0078   KEXAM  = 003E
KEYBRD= 00C8   KEY1   = 0078   KEY10  = 006D   KEY11  = 006E
```

```
KEY12 = 006F    KEY13 = 005C    KEY14 = 005D    KEY15 = 005E
KEY16 = 005F    KEY17 = 003C    KEY18 = 003D    KEY19 = 003E
KEY2  = 0079    KEY20 = 003F    KEY3  = 007A    KEY4  = 007B
KEY5  = 0074    KEY6  = 0075    KEY7  = 0076    KEY8  = 0077
KEY9  = 006C    KINTA = 006F    KLDAD = 003D    KN0   = 003C
KN1   = 005C    KN2   = 005D    KN3   = 005E    KN4   = 006C
KN5   = 006D    KN6   = 006E    KN7   = 0074    KN8   = 0075
KN9   = 0076    KU2   = 0079    KU3   = 007A    KU8   = 0077
L     =%0005    LBLANK= 000F    LCE   = 000B    LCH   = 000C
LCL   = 000D    LCP   = 000E    LC0   = 0000    LC1   = 0001
LC2   = 0002    LC3   = 0003    LC4   = 0004    LC5   = 0005
LC6   = 0006    LC7   = 0007    LC8   = 0008    LC9   = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94
LPNUM   4F92    M     =%0006    MBSEL = 00E0    MB.A  = 0008
MB.B  - 0004    MEMTOP= 4FFF    MINUS = 000A    MM    = 8000
MSE   = 0008    MSGN  = 000F    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV- 0020
MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT= 0080    M.DEM = 0020    M.EBL = 0010    M.EXC = 0008
M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010    M.OCC = 0020
M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5  = 0001    M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088
OKAY  = 00FF    OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040
PADCNT= 00D5    PADCRC= 0080    PDIAG = 0048    PEID  = 008A
PENAB = 004C    PESET = 0001    PL    = 00B1    POINTB  4870
POINTE  4858    P04X5   47DC    PRDD  = 004C    PRENF = 009C
PS    = 00B2    PSTAT = 0048    PSW   =%0009    P.AMTP= 0001
P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040
P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002
P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010
P.SING= 0080    P.STAT= 0002    P.STPE= 0080    P.TACH= 0008
P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001
P.WPEN= 0010    P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008
P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B    QUEM  = 281E
RAMT  - 0010    RARA  = 0006    RARAI = 0004    RCHBD0= 0048
RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C    RCLRT = 000D
RCMD  = 0008    RCMLP = 0003    RCONT = 0080    RDATA = 0017
RDCLK = 0010    RDON  = 0011    READG = 0007    REND  = 0014
REQST = 2806    RESCHR= 00D1    RETRY   4777    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 000C    RMCTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 000C
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
```

```
R.DATA= 0040    R.DON = 0002    R.DRDY= 0010    R.END = 0010
R.ILL = 0004    R.JVOK= 0004    R.MK2 = 0008    R.PLOD= 0008
R.PLO0= 0010    R.PLO1= 0020    R.POST= 0020    R.STNM= 0002
R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080    R.TSTD= 0040
R.VOK = 0080    R00H  = 0081    R00L  = 0080    R01H  = 0083
R01L  = 0082    R02H  = 0085    R02L  = 0084    R03H  = 0087
R03L  = 0086    R04H  = 0089    R04L  = 0088    R05H  = 008B
R05L  = 008A    R06H  = 008D    R06L  = 008C    R07H  = 008F
R07L  = 008E    R10H  = 0091    R10L  = 0090    R11H  = 0093
R11L  = 0092    R12H  = 0095    R12L  = 0094    R13H  = 0097
R13L  = 0096    R14H  = 0099    R14L  = 0098    R15H  = 009B
R15L  = 009A    R16H  = 009D    R16L  = 009C    R17H  = 009F
R17L  = 009E    R7.5  = 0010    SELCLR= 0000    SETATA= 00A1
SID   = 0080    SOD   = 0080    SOE   = 0040    SP    =%0008
SSCLK = 0040    SSTEP = 0005    STACK = 4FFF    STATRM= 4F20
STPCT   4F20    STRSP = 5000    SUNIT   4935    TAB4X5  4874
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = C044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TEST2   4543    TIET1   4932
TIET2   4933    TIET3   4934    TIMTBL  4929    TMF   = 0099
TMRDY = 004^    TRANS   48C6    TRANS1  48C8    TRANS2  48D0
TRK     48D     TRKA    48E4    TRKB    48FF    TR<C    4919
TRKENA= 00D2    TRNIN   4884    TRNOUT  4888    TRNTMP  4893
TSET  = 2803    TSTEND= 2818    TSTS  = 0040    TST01A  4430
TST01B  4434    TST01C  437F    TST01F  4397    TST01G  43C4
TST01H  43F6    TST01I  43A3    TST01K  4483    TST01L  4496
TST01M  44AA  G TST01N  44EC    TST01Q  44D1    TST01R  44C8  G
TST01S  44E1    TST01T  44DB  G TST01U  4503  G TST01V  451F  G
TST01X  4351    TST01Y  4485    TST02A  464C    TST02B  4653
TST02C  457C    TST02F  4597    TST02G  45E1    TST02H  4612
TST02I  45AA    TST02K  46B2    TST02L  46C6    TST02M  46D9  G
TST02N  471F    TST02Q  4700    TST02R  46F7  G TST02S  4714
TST02T  470E  G TST02U  4736  G TST02V  4752  G TST02W  45B6
TST02X  454D    TST02Y  46B7    TST02Z  45BD    TUSEL0= 00D1
TUSEL1= 00D2    TU78  = 0010    T.ATH0= 0001    T.ATH1= 0002
T.BOT = 0004    T.EOT = 0002    T.FPT = 0001    T.NTHR= 0004
T.ONL = 0020    T.PES = 0008    T.PSBJ= 0020    T.PSOJ= 0008
T.PS1J= 0010    T.RDY = 0080    T.RDYO= 0040    T.RWD = 0010
T.SCLK= 0002    T4X5    47B1    UIBG  = 00A1    UNITMP  4938
UNITSL  4327    VALFC   4F98    VALTB   4F95    VELTST= 005B
WDR.P = 0010    WMCCTL= 00D3    WMCERR= 00DA    WMCSTA= 00D0
WRTCLK= 0000    WRTDAT= 00D3    W.ACRC= 0004    W.CRC = 0008
W.DIAG= 0002    W.DONN= 0040    W.ECC = 0010    W.ENAB= 0080
W.ERR = 0020    W.FMT = 0070    W.GCR = 0010    W.LEFT= 0004
W.ONES= 0020    W.RESI= 0002    W.REV = 0004    W.ROME= 0010
W.RST = 0001    W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020
X     =%000A    X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002
X.ROME= 0001    X.WCLK= 0001    Y     =%0008    .     = 4946
```

```
 ERRORS DETECTED:  0

 *ECC4.A78/PTP,ECC4=NLIST,PARAM,MACRO,LIST,ECC4
 RUN-TIME: 4 7 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:06 SEQ 1017
PARAM.M80 TABLE OF CONTENTS

```
1329                                .TITLE  ECC5 - ECC CONTROLLER PART #5
1330                                .SBTTL  TEST 1 - ECC BAD DATA TWO TRACKS TWO HARD AND SEVEN SOFT POINTERS
1331                                ;ID     ECC5-ERROR CORRECTION CONTROLLER PART #5
1332  4300                          ST
 (1)                                ;*********************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*----- -----
1333                                ;*ECC BAD DATA TWO TRACKS TWO HARD AND SEVEN SOFT POINTERS
1334  4300                          SD
 (1)                                ;*********************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1335                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM TWO
1336                                ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN TWO TRACKS, AND TWO
1337                                ;*CORRECT POINTERS ALSO PROVIDED. HOWEVER, AN ADDITIONAL 7 SOFT POINTERS
1338                                ;*ARE PROVIDED TO TEST THE ABILITY OF THE ECC TO DESCRIMINATE
1339                                ;*BETWEEN HARD AND SOFT POINTERS.  THE PROGRAM CHECKS THAT THE DATA IS
1340                                ;*CORRECTED, AND THAT THE TWO TRACK ERROR CORRECT STATUS BIT IS SET.
1341                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
1342                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
1343                                ;*MESSAGE, AND THE DATA BIT POSITIONS OF THE INCORRECT DATA
1344                                ;*CAN BE DETERMINED FROM THE FOLLOWING TABLE.
1345                                ;*
1346                                ;*                                                  DATA BIT POSITION
1347                                ;*      BYTE COUNT              DATA                WITH HARD POINTERS
1348                                ;*      ----------              ----                ------------------
1349                                ;*
1350                                ;*        001          000,001,002,003,004,005,006        7,P
1351                                ;*        002          007,010,011,012,013,014,015        6,7
1352                                ;*        003          016,017,020,021,022,023,024        5,6
1353                                ;*        004          025,026,027,030,031,032,033        4,5
1354                                ;*        005          034,035,036,037,040,041,042        3,4
1355                                ;*        006          043,044,045,046,047,050,051        2,3
1356                                ;*        007          052,053,054,055,056,057,060        1,2
1357                                ;*        010          061,062,063,064,065,066,067        0,0
1358                                ;*        011          070,071,072,073,074,075,076        7,P
1359                                ;*        012          077,100,101,102,103,104,105        6,7
1360                                ;*        013          106,107,110,111,112,113,114        5,6
1361                                ;*        014          115,116,117,120,121,122,123        4,5
1362                                ;*        015          124,125,126,127,130,131,132        3,4
1363                                ;*        016          133,134,135,136,137,140,141        2,3
1364                                ;*        017          142,143,144,145,146,147,150        1,2
1365                                ;*        020          151,152,153,154,155,156,157        0,1
1366                                ;*        021          160,161,162,163,164,165,166        7,P
1367                                ;*        022          167,170,171,172,173,174,175        6,7
1368                                ;*        023          176,177,200,201,202,203,204        5,6
1369                                ;*        024          205,206,207,210,211,212,213        4,5
1370                                ;*        025          214,215,216,217,220,221,222        3,4
1371                                ;*        026          223,224,225,226,227,230,231        2,3
1372                                ;*        027          232,233,234,235,236,237,240        1,2
1373                                ;*        030          241,242,243,244,245,246,247        P,0
1374                                ;*        031          250,251,252,253,254,255,256        7,P
1375                                ;*        032          257,260,261,262,263,264,265        6,7
1376                                ;*        033          266,267,270,271,272,273,274        5,6
```

```
                        ;*          034          275,276,277,300,301,302,303          4,5
                        ;*          035          304,305,306,307,310,311,312          3,4
                        ;*          036          313,314,315,316,317,320,321          2,3
                        ;*          037          322,323,324,325,326,327,330          1,2
                        ;*          040          331,332,333,334,335,336,337          0,1
                        ;*          041          340,341,342,343,344,345,346          7,P
                        ;*          042          347,350,351,352,353,354,355          6,7
                        ;*          043          356,357,360,361,362,363,364          5,6
                        ;*          044          365,366,367,370,371,372,373          4,5
                        ;*          045          374,375,376,377,000,001,002          3,4
                        ;*
                        ;*THE POINTERS PROVIDED FOR THE TWO TRACKS IN ERROR ARE BAD 4X5.  THE
                        ;*POINTERS PROVIDED FOR THE OTHER POSITIONS ARE PHASE TIE.
4300                    SP
 (1)                    ; ***************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
                        ;*BGNTST
                        ;*    SET NORMAL READ PATH CLOCK
                        ;*    CALL SUBROUTINE CLEAR
                        ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                        ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                        ;*    DELAY 50 MICROSECONDS
                        ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                        ;*    CLEAR THE INPUT DATA
                        ;*    CLEAR BAD TRANSLATION POINTERS
                        ;*  BGNDO
                        ;*  : ENABLE THE READ PATH CLOCK
                        ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                        ;*  : CLOCK THE FIFO'S
                        ;*  : ISSUE CLEAR ALL COMMAND
                        ;*  : WAIT
                        ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                        ;*  : CALL SUBROUTINE CLECC
                        ;*  : SET THE LOOP COUNT TO 7
                        ;*  : BGNDO
                        ;*  : :  GET THE INPUT DATA BYTE
                        ;*  : :  STORE IN THE BUFFER
                        ;*  : :  CALL SUBROUTINE ECC
                        ;*  : :  DECREMENT THE LOOP COUNT
                        ;*  : :  INCREMENT THE INPUT DATA
                        ;*  : :  DO UNTIL THE LOOP COUNT 0
                        ;*  : ENDDO
                        ;*  : SET THE LOOP COUNT TO 3
                        ;*  : BGNDO
                        ;*  : :  FILL THE TRANSLATOR INPUT BUFFER WITH THE DATA FROM THE BUFFER
                        ;*  : :  CALL SUBROUTINE TRK
                        ;*  : :  CALL SUBROUTINE BADT4X5
                        ;*  : :  INIT THE LOOP COUNT TO 5
                        ;*  : :  BGNDO
                        ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                        ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                        ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                        ;*  : :  :  CLOCK THE DATA INTO THE FIFO
```

```
1428                                    ;*  : : :  DECREMENT THE LOOP COUNT
1429                                    ;*  : : :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1430                                    ;*  : : :  DO UNTIL LOOP COUNT = 0
1431                                    ;*  : : ENDDO
1432                                    ;*  : : FILL THREE BYTES OF TRANSLATOR INPUT BUFFER WITH THE BUFFER DATA
1433                                    ;*  : : FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1434                                    ;*  : : CALL SUBROUTINE TRK
1435                                    ;*  : : CALL SUBROUTINE BADT4X5
1436                                    ;*  : : INIT THE LOOP COUNT TO 5
1437                                    ;*  : : BGNDO
1438                                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1439                                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1440                                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1441                                    ;*  : : :  DECREMENT THE LOOP COUNT
1442                                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1443                                    ;*  : : ENDDO
1444                                    ;*  : : DECREMENT THE LOOP COUNT
1445                                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1446                                    ;*  : ENDDO
1447                                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1448                                    ;*  : SET UP WATCHDOG TIMER COUNT
1449                                    ;*  : BGNDO
1450                                    ;*  : :    SINGLE STEP THE READ PATH
1451                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1452                                    ;*  : :    DO UNTIL WATCHDOG TIMER - 0 OR DATA READY SETS
1453                                    ;*  : ENDDO
1454                                    ;*  : IF WATCHDOG TIMER=0
1455                                    ;*  : :    THEN-ERROR
1456                                    ;*  : :    ELSE-CONTINUE
1457                                    ;*  : ENDIF
1458                                    ;*  : GET THE TIE BUS TABLE ADDRESS
1459                                    ;*  : SINGLE STEP THE READ PATH
1460                                    ;*  : INIT THE LOOP COUNT TO 0
1461                                    ;*  : BGNDO
1462                                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1 OR JAM COUNT 2
1463                                    ;*  : :    :  THEN-LOAD A BAD 4X5 POINTER
1464                                    ;*  : :    :  ELSE-LOAD A PHASE TIE POINTER
1465                                    ;*  : :    ENDIF
1466                                    ;*  : :    SINGLE STEP THE READ PATH
1467                                    ;*  : :    DECREMENT THE LOOP COUNT
1468                                    ;*  : :    DO UNTIL LOOP COUNT = 9
1469                                    ;*  : ENDDO
1470                                    ;*  : SET UP WATCHDOG TIMER COUNT
1471                                    ;*  : BGNDO
1472                                    ;*  : :  SINGLE STEP THE READ PATH
1473                                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1474                                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1475                                    ;*  : ENDDO
1476                                    ;*  : IF WATCHDOG TIMER=0
1477                                    ;*  : :  THEN-ERROR-EXIT TEST
1478                                    ;*  : :  ELSE-CONTINUE
1479                                    ;*  : ENDIF
1480                                    ;*  : SINGLE STEP THE READ PATH
1481                                    ;*  : INIT THE  LOOP COUNT TO 7
```

```
                                ;*   : BGNDO
                                ;*   : :  SINGLE STEP THE READ PATH
                                ;*   : :  COMPARE CORRECTED DATA WITH INPUT DATA
                                ;*   : :  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
                                ;*   : :  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  DECREMENT THE LOOP COUNT
                                ;*   : :  DO UNTIL THE LOOP COUNT - 0
                                ;*   : ENDDO
                                ;*   : SINGLE STEP THE READ PATH
                                ;*   : IF EXPECTED NOT EQUAL
                                ;*   : :  THEN-ECC ERROR
                                ;*   : :  ELSE-CONTINUE
                                ;*   : ENDIF
                                ;*   : INPUT THE ECC STATUS
                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*   : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
                                ;*   : :  THEN-CONTINUE
                                ;*   : :  ELSE-ERROR
                                ;*   : ENDIF
                                ;*   : INCREMENT THE INPUT DATA
                                ;*   : DO UNTIL INPUT DATA = 0
                                ;*   ENDDO
                                ;*ENDTST
1512  4300                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1513                            ;*ECC5 MICRO TEST 01
1514                            ;*ECC5 MICRO ERROR 01
1515                            ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1516                            ;*M8951, M8950'S
1517                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1518                            ;*FATAL ERROR - TEST ABORTED
1519                            ;*
1520                            ;*ECC5 MICRO TEST 01
1521                            ;*ECC5 MICRO ERROR 02
1522                            ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1523                            ;*M8951, M8950'S
1524                            ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1525                            ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1526                            ;*'BYTE/SCLK COUNT NUMBER = LLL''
1527                            ;*FATAL ERROR - MICRO TEST ABORTED
1528                            ;*
1529                            ;*ECC5 MICRO TEST 01
1530                            ;*ECC5 MICRO ERROR 03
1531                            ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1532                            ;*M8951, M8950'S
```

```
                                        ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                                        ;*SENT TO THE ECC
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*FATAL ERROR - MICRO TEST ABORTED
                                        ;*
                                        ;*ECC5 MICRO TEST 01
                                        ;*ECC5 MICRO ERROR 04
                                        ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA FROM THE ECC-INCORRECT
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC5 MICRO TEST 01
                                        ;*ECC5 MICRO ERROR 05
                                        ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ZERO SHOULD - ONE
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC5 MICRO TEST 01
                                        ;*ECC5 MICRO ERROR 06
                                        ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC5 MICRO TEST 01
                                        ;*ECC5 MICRO ERROR 07
                                        ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC5 MICRO TEST 01
                                        ;*ECC5 MICRO ERROR 10
                                        ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        S
 4300                                   ; ****** ********************************************************************************
  (1)
                                        TEST1:  TEST  @1                ;INITIALIZE THE TEST
 4300  3E  01              7.0                         MVI     A,@1            ;DEFINE THE TEST NUMBER
 4302  CD  03  28          18.0                        CALL    TSET            ;SETUP THE TEST
                                        ;%ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;&M8951, M8950'S

                                                REQ     @7,0,0,0,0
 4305  CD  06  28          18.0                        CALL    REQST
```

```
 (1) 4308 00                                        .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4309 00 00                                     .WORD   0               ;SYSTEM "0" COUNT
 (1) 430B 00 00                                     .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D 00                                        .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1) 430E 07                                        .BYTE   @7              ;REQUEST CODE
1583 430F                           RIN     R12L
 (1) 430F DB 94          10.0                       IN      R12L            ;READ R12L INTO AC
 (1) 4311 7F              4.0                       MOV     A,A             ;RETRY LINK
1584 4312 32 56 49       13.0       STA     UNITMP
1585 4315 A7              4.0       ANA     A
1586 4316 C2 27 43       10.0       JNZ     UNITSL
1587 4319                           ERR     EXIT,DUMMY
 (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 4319 CD 09 28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      0001                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 431C 01                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 431D 00                                        .BYTE
 (1) 431E CD 15 28        18.0               DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4321 DA 95 47        10.0                       JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1588                            ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1589                            ;>FATAL ERROR - TEST ABORTED
1590 4324 C3 95 47       10.0       JMP     EXIT
1591
1592 4327 06 00           7.0 UNITSL: MVI     B,0
1593 4329 3A 56 49       13.0       LDA     UNITMP          ;DID THE USER SPECIFY TU
1594 432C E6 01           7.0       ANI     @01             ;PORT 0?
1595 432E C2 44 43       10.0       JNZ     FOUND           ;YES-GO USE PORT #0
1596 4331 04              4.0       INR     B               ;NO-UPDATE POINTER TO PORT #1
1597 4332 3A 56 49       13.0       LDA     UNITMP          ;DID THE USER SPECIFY TU
1598 4335 E6 02           7.0       ANI     @02             ;PORT 1?
1599 4337 C2 44 43       10.0       JNZ     FOUND           ;YES-GO USE PORT #1
1600 433A 04              4.0       INR     B               ;NO-UPDATE POINTER TO PORT #2
1601 433B 3A 56 49       13.0       LDA     UNITMP          ;DID THE USER SPECIFY TU
1602 433E E6 04           7.0       ANI     @04             ;PORT 2?
1603 4340 C2 44 43       10.0       JNZ     FOUND           ;YES-GO USE PORT #2
1604 4343 04              4.0       INR     B               ;NO-ASSUME PORT #3
1605 4344 CD A3 47       18.0 FOUND: CALL    CLEAR           ;CLEAR ALL TU PORTS
1606 4347 DB E0          10.0       IN      INTSTA
1607 4349 E6 80           7.0       ANI     BIT7
1608 434B B0              4.0       ORA     B
1609 434C D3 E0          10.0       OUT     MBSEL
1610 434E 32 53 49       13.0       STA     SUNIT           ;SAVE FOR NEXT TEST
1611
1612 4351 3E 10           7.0 TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1613 4353 D3 F0          10.0       OUT     CLKCTL
1614 4355 3E 10           7.0       MVI     A,W.GCR
1615 4357 D3 D3          10.0       OUT     WMCCTL
1616 4359 3E 04           7.0       MVI     A,P.RPEN
1617 435B D3 4C          10.0       OUT     PENAB
1618
1619 435D 3E 60           7.0       MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1620 435F D3 48          10.0       OUT     PDIAG           ;LCS, LWR
1621 4361 AF              4.0       XRA     A
1622 4362 D3 44          10.0       OUT     TAMT
```

```
1623  4364  3E  0A        7.0          MVI   A,10              ;DELAY
1624  4366  3D            4.0  1$:     DCR   A                 ;50
1625  4367  C2  66  43    10.0         JNZ   1$                ;MICROSECONDS
1626
1627  436A  D3  D2        10.0         OUT   TRKENA            ;CLEAR ALL TRACKS FROM
1628  436C  D3  0A        10.0         OUT   RTIEB             ;CLEAR THE TIE BUS REGISTER
1629  436E  D3  D2        10.0         OUT   TRKENA
1630  4370  32  B1  48    13.0         STA   BAD1              ;CLEAR THE INVALID
1631  4373  32  B2  48    13.0         STA   BAD2              ;TRACK TRANSLATIONS
1632  4376  32  B3  48    13.0         STA   BAD3              ;POINTERS
1633  4379  32  54  49    13.0         STA   INDAT             ;CLEAR THE INPUT DATA BYTE
1634  437C                             ROUT  R05H
 (1)  437C  D3  8B        10.0                OUT   R05H              ;WRITE AC INTO R05H
 (1)  437E  7F            4.0                 MOV   A,A               ;RETRY LINK
1635
1636  437F  3E  A8        7.0  TST01C: MVI   A,R.PL01.R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1637                                                                  ;SET TIE BUS JAM AND PLO DISABLE
1638  4381  D3  09        10.0         OUT   RPCTL
1639  4383  D3  08        10.0         OUT   RFIFOL            ;CLOCK THE FIFO'S
1640  4385  3E  0D        7.0          MVI   A,RCLRT           ;ISSUE CLEAR ALL COMMAND
1641  4387  D3  0B        10.0         OUT   RCMD
1642  4389  00            4.0          NOP                     ;WAIT
1643  438A  00            4.0          NOP
1644  438B  00            4.0          NOP
1645  438C  00            4.0          NOP
1646  438D  00            4.0          NOP
1647  438E  3E  A9        7.0          MVI   A,R.PL01!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1648                                                                  ;SET TIE BUS JAM AND PLO DISABLE
1649  4390  D3  09        10.0         OUT   RPCTL
1650  4392  CD  B4  48    18.0         CALL  CLECC             ;CLEAR THE ECC CHARACTER
1651  4395  21  5D  49    10.0         LXI   H,BUFFER          ;SET UP THE BUFFER POINTER
1652  4398  16  07        7.0          MVI   D,@7              ;SET UP THE LOOP COUNT
1653  439A  3A  54  49    13.0 TST01F: LDA   INDAT             ;GET THE INPUT DATA
1654  439D  77            7.0          MOV   M,A
1655  439E  3C            4.0          INR   A
1656  439F  32  54  49    13.0         STA   INDAT
1657  43A2  23            6.0          INX   H
1658  43A3  3D            4.0          DCR   A                 ;RECREATE THE ORIGINAL DATA FOR THE ECC SUB
1659  43A4  CD  B9  48    18.0         CALL  ECC               ;CALCULATE ECC FOR
1660  43A7  15            4.0          DCR   D                 ;THE 7 INPUT BYTES
1661  43A8  C2  9A  43    10.0         JNZ   TST01F
1662  43AB  0E  03        7.0          MVI   C,@3              ;INIT THE LOOP COUNT
1663  43AD  3A  54  49    13.0 TST01I: LDA   INDAT             ;GET THE INPUT DATA
1664  43B0  06  25        7.0          MVI   B,@45
1665  43B2  FE  03        7.0          CPI   @3
1666  43B4  CA  C0  43    10.0         JZ    TST01Z
1667  43B7  06  00        7.0          MVI   B,0               ;CLEAR THE BYTE COUNT
1668  43B9  04            4.0  TST01W: INR   B
1669  43BA  D6  07        7.0          SUI   @7                ;DIVIDE BY 7
1670  43BC  A7            4.0          ANA   A
1671  43BD  C2  B9  43    10.0         JNZ   TST01W
1672  43C0  78            4.0  TST01Z: MOV   A,B
1673  43C1                             ROUT  R05L
 (1)  43C1  D3  8A        10.0                OUT   R05L              ;WRITE AC INTO R05L
```

```
  (1)  43C3  7F          4.0                MOV     A,A             ;RETRY LINK
1674   43C4  3A 5D 49     13.0       LDA     BUFFER
1675   43C7  32 A1 48     13.0       STA     TRNIN           ;FILL THE TRANSLATOR
1676   43CA  3A 5E 49     13.0       LDA     BUFFER+1
1677   43CD  32 A2 48     13.0       STA     TRNIN+1         ;SUBROUTINE INPUT
1678   43D0  3A 5F 49     13.0       LDA     BUFFER+2
1679   43D3  32 A3 48     13.0       STA     TRNIN+2         ;BUFFER
1680   43D6  3A 60 49     13.0       LDA     BUFFER+3
1681   43D9  32 A4 48     13.0       STA     TRNIN+3         ;
1682   43DC  CD F5 48     18.0       CALL    TRK
1683   43DF  AF            4.0       XRA     A               ;CLEAR THE ACCUMULATOR
1684   43E0  32 B3 48     13.0       STA     BAD3            ;CLEAR THE THIRD POINTER
1685   43E3  CD CE 47     18.0       CALL    BADT4X5         ;TRANSLATE THE SUBGROUP
1686   43E6  21 A5 48     10.0       LXI     H,TRNOUT        ;GET POINTER TO TRANSLATE DATA TABLE
1687   43E9  06 05         7.0       MVI     B,@5            ;SET UP LOOP COUNT
1688   43EB  7E            7.0 TST01G: MOV    A,M             ;GET A DATA BYTE
1689   43EC  D3 40        10.0       OUT     TCMD            ;STORE DATA IN COMMAND ADDRESS
1690   43EE  23            6.0       INX     H               ;POINT TO DATA PARITY
1691   43EF  7E            7.0       MOV     A,M             ;GET THE DATA PARITY
1692   43F0  07            4.0       RLC                     ;POSITION FOR OUTPUT
1693   43F1  F6 60         7.0       ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
1694   43F3  D3 48        10.0       OUT     PDIAG           ;OUTPUT THE DATA PARITY
1695   43F5  D3 08        10.0       OUT     RFIFOL          ;CLOCK DATA INTO THE FIFO'S
1696   43F7  23            6.0       INX     H               ;UPDATE THE TABLE POINTER
1697   43F8  05            4.0       DCR     B               ;DECREMENT LOOP COUNT
1698   43F9  C2 EB 43     10.0       JNZ     TST01G          ;DO UNTIL LOOP COUNT = 0
1699   43FC  3A 61 49     13.0       LDA     BUFFER+4        ;GET THE INPUT DATA
1700   43FF  32 A1 48     13.0       STA     TRNIN           ;FILL THE FIRST THREE
1701   4402  3A 62 49     13.0       LDA     BUFFER+5
1702   4405  32 A2 48     13.0       STA     TRNIN+1         ;BYTES OF THE TRANSLATOR
1703   4408  3A 63 49     13.0       LDA     BUFFER+6
1704   440B  32 A3 48     13.0       STA     TRNIN+2         ;SUBROUTINE WITH THE INPUT DATA
1705   440E  3A E2 48     13.0       LDA     ECCCHR          ;STORE THE CALCULATED
1706   4411  32 A4 48     13.0       STA     TRNIN+3         ;ECC CHARACTER AS THE LAST CHARACTER
1707   4414  CD F5 48     18.0       CALL    TRK
1708   4417  AF            4.0       XRA     A               ;CLEAR THE ACCUMULATOR
1709   4418  32 B3 48     13.0       STA     BAD3            ;CLEAR THE THIRD POINTER
1710   441B  CD CE 47     18.0       CALL    BADT4X5         ;TRANSLATE THE SECOND SUBGROUP
1711   441E  21 A5 48     10.0       LXI     H,TRNOUT        ;GET THE POINTER TO INPUT DATA TABLE
1712   4421  06 05         7.0       MVI     B,@5            ;SET UP THE LOOP COUNT
1713   4423  7E            7.0 TST01H: MOV    A,M             ;GET A DATA BYTE
1714   4424  D3 40        10.0       OUT     TCMD            ;STORE DATA IN CMD ADDRESS
1715   4426  23            6.0       INX     H               ;POINT TO DATA PARITY BIT
1716   4427  7E            7.0       MOV     A,M             ;GET THE DATA PARITY
1717   4428  07            4.0       RLC                     ;POSITION FOR OUTPUT
1718   4429  F6 60         7.0       ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
1719   442B  D3 48        10.0       OUT     PDIAG           ;OUTPUT THE DATA PARITY
1720   442D  D3 08        10.0       OUT     RFIFOL          ;CLOCK DATA INTO THE FIFOS
1721   442F  23            6.0       INX     H               ;UPDATE THE TABLE POINTER
1722   4430  05            4.0       DCR     B               ;DECREMENT THE LOOP COUNT
1723   4431  C2 23 44     10.0       JNZ     TST01H          ;DO UNTIL LOOP COUNT=0
1724   4434  0D            4.0       DCR     C               ;PUT THE GROUP IN
1725   4435  C2 AD 43     10.0       JNZ     TST01I          ;THE FIFO TWICE
1726   4438  3E 0B         7.0       MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
```

```
1727  443A  D3  0B        10.0          OUT   RCMD              ;COMMAND
1728  443C  11  01  00     10.0          LXI   D,1               ;SET WATCH DOG INCREMENT
1729  443F  21  A8  FD     10.0          LXI   H,-600            ;SET WATCH DOG COUNT TO 600
1730  4442  D3  0C        10.0  2$:     OUT   RINST             ;STEP THE READ PATH
1731  4444  DB  01        10.0          IN    RPCHI             ;DATA READY SET?
1732  4446  E6  10         7.0          ANI   R.DRDY
1733  4448  C2  5D  44     10.0          JNZ   TST01A            ;YES-GO PROCESS
1734  444B  19             10.0          DAD   D                 ;WATCH DOG TIMEOUT?
1735  444C  D2  42  44     10.0          JNC   2$                ;NO-CONTINUE
1736  444F                               ERR   TST01C,DUMMW,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  444F  CD  09  28     18.0                CALL  ERLP              ;PROCESS ERROR - DO 2.3
 (1)        0002                           MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4452  02                             .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  4453  01                             .BYTE 1
 (1)  4454  CD  15  28     18.0    DUMMW:: CALL  CKLOP             ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4457  DA  7F  43     10.0          JC    TST01C            ;LOOP ADDRESS IF LOOP SPECIFIED
1737                                ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1738                                ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1739                                ;<FATAL ERROR - MICRO TEST ABORTED
1740  445A  C3  95  47     10.0          JMP   EXIT
1741
1742  445D  D3  0C        10.0  TST01A: OUT   RINST
1743  445F  06  00         7.0          MVI   B,$0              ;INIT THE BIT TIME COUNT (LOOP COUNT)
1744  4461  00             4.0  TST01B: NOP
1745  4462  3A  50  49     13.0          LDA   TIET1             ;GET FIRST TIE BUS JAM TIME
1746  4465  B8             4.0          CMP   B                 ;IS IT TIME TO JAM THE TIE BUS?
1747  4466  C2  70  44     10.0          JNZ   4$                ;NO - CONTINUE
1748  4469  3E  08         7.0          MVI   A,$8              ;LOAD A BAD
1749  446B  D3  0A        10.0          OUT   RTIEB             ;4X5 POINTER
1750  446D  C3  82  44     10.0          JMP   5$                ;
1751  4470  3A  51  49     13.0  4$:     LDA   TIET2             ;GET SECOND TIE BUS JAM TIME
1752  4473  B8             4.0          CMP   B                 ;IS IT TIME TO JAM THE TIE BUS?
1753  4474  C2  7E  44     10.0          JNZ   6$                ;NO-CONTINUE
1754  4477  3E  08         7.0          MVI   A,$8              ;LOAD A BAD
1755  4479  D3  0A        10.0          OUT   RTIEB             ;4X5 POINTER
1756  447B  C3  82  44     10.0          JMP   5$                ;
1757  447E  3E  02         7.0  6$:     MVI   A,@2              ;IF NOT THE BAD TRACK, PROVIDE A PHASE TIE POINTER
1758  4480  D3  0A        10.0          OUT   RTIEB             ;LOAD THE POINTER
1759  4482  D3  0C        10.0  5$:     OUT   RINST             ;STEP CLOCK
1760  4484  04             4.0          INR   B                 ;DECREMENT THE LOOP COUNT
1761  4485  78             4.0          MOV   A,B
1762  4486  FE  09         7.0          CPI   $9                ;DONE
1763  4488  C2  61  44     10.0          JNZ   TST01B            ;DO UNTIL LOOP COUNT = ZERO
1764  448B  AF             4.0          XRA   A
1765  448C  D3  0A        10.0          OUT   RTIEB             ;CLEAR THE TIE BUS REGISTER
1766  448E  D3  0C        10.0          OUT   RINST             ;CLOCK THE READ PATH
1767  4490  D3  0C        10.0          OUT   RINST             ;CLOCK THE READ PATH
1768  4492  11  01  00     10.0          LXI   D,1               ;SET WATCHDOG TIMER INCREMENT
1769  4495  21  A8  FD     10.0          LXI   H,-600            ;SET WATCHDOG COUNT TO 600
1770  4498  D3  0C        10.0  3$:     OUT   RINST             ;SINGLE STEP THE READ PATH
1771  449A  DB  01        10.0          IN    RPCHI             ;DATA READY SET?
1772  449C  E6  10         7.0          ANI   R.DRDY
1773  449E  C2  B3  44     10.0          JNZ   TST01K            ;YES-GO PROCESS
```

```
1774  44A1  19                10.0          DAD     D                ;NO-WATCHDOG TIMEOUT?
1775  44A2  D2  98  44         10.0          JNC     3$               ;NO-CONTINUE
1776  44A5                                   ERR     TST01C,DUMMX,1
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44A5  CD  09  28         18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0003                                     MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A8  03                                       .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44A9  01                                       .BYTE   1
 (1)  44AA  CD  15  28         18.0          DUMMX:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44AD  DA  7F  43         10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1777                                   ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1778                                   ;>SENT TO THE ECC
1779                                   ;<FATAL ERROR - MICRO TEST ABORTED
1780  44B0  C3  95  47         10.0          JMP     EXIT
1781  44B3  16  01               7.0  TST01K: MVI     D,@1             ;INIT THE LOOP COUNT
1782  44B5  21  5D  49         10.0          LXI     H,BUFFER
1783  44B8  00                    4.0  TST01Y: NOP                      ;
1784  44B9  7E                    7.0          MOV     A,M              ;GET THE INPUT DATA
1785  44BA                                   ROUT    EDATA
 (1)  44BA  D3  95               10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  44BC  7F                    4.0                  MOV     A,A              ;RETRY LINK
1786  44BD  A7                    4.0          ANA     A                ;SET CONDITION BITS
1787  44BE  3E  00                7.0          MVI     A,0              ;CLEAR THE ACC.
1788  44C0  E2  C4  44          10.0          JPO     1$               ;ODD # OF ONES - CONTINUE
1789  44C3  3C                    4.0          INR     A                ;EVEN # OF ONES - SET PARITY
1790  44C4  32  55  49          13.0  1$:     STA     INDATP           ;STORE THE PARITY BIT
1791  44C7  D3  0C               10.0  TST01L: OUT     RINST            ;SINGLE STEP THE READ LOGIC
1792  44C9  7E                    7.0          MOV     A,M
1793  44CA  23                    6.0          INX     H
1794  44CB  47                    4.0          MOV     B,A
1795  44CC  DB  19               10.0          IN      ECCCOR
1796  44CE  B8                    4.0          CMP     B
1797  44CF  CA  DA  44          10.0          JZ      TST01M
1798  44D2                                   ROUT    ADATA
 (1)  44D2  D3  94               10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  44D4  7F                    4.0                  MOV     A,A              ;RETRY LINK
1799  44D5                                   ERRB    TST01C,TST01M,@1
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  44D5  CD  12  28         18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0004                                     MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D8  04                                       .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44D9  01                                       .BYTE   @1               ;PRINT ROUTINE NUMBER
 (1)  44DA  CD  15  28         18.0          TST01M::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44DD  DA  7F  43         10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1800                                   ;>CORRECTED DATA FROM THE ECC-INCORRECT
1801  44E0  DB  15               10.0          IN      RPSTA            ;GET THE ECC PARITY BIT
1802  44E2  E6  80                7.0          ANI     E.CDP            ;GET THE CORRECTED PARITY BIT
1803  44E4  07                    4.0          RLC                      ;POSITION BIT FOR COMPARE
1804  44E5  4F                    4.0          MOV     C,A              ;SAVE IN REG. C
1805  44E6  3A  55  49          13.0          LDA     INDATP           ;GET THE EXPECTED PARITY BIT
1806  44E9  A7                    4.0          ANA     A                ;SET THE CONDITION BITS
1807  44EA  CA  01  45          10.0          JZ      TST01QO          ;GO CHECK FOR PARITY=0
1808  44ED  79                    4.0          MOV     A,C              ;ELSE CHECK FOR PARITY=1
1809  44EE  FE  01                7.0          CPI     @1
```

```
1810  44F0  CA  F8  44      10.0          JZ      TST01RG
1811  44F3                               ERR     TST01C,TST01RG,@1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F3  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                          MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F6  05                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F7  01                            .BYTE   @1
 (1)  44F8  CD  15  28      18.0          TST01RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FB  DA  7F  43      10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1812                                ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1813  44FE  C3  11  45      10.0          JMP     TST01SZ                 ;CONTINUE WITH TEST
1814  4501  79                 4.0    TST01QO: MOV    A,C               ;CHECK FOR PARITY = ZERO
1815  4502  A7                 4.0          ANA     A                   ;SET CONDITION BITS
1816  4503  CA  0B  45      10.0          JZ      TST01TP                 ;CONTINUE IF ZERO
1817  4506                               ERR     TST01C,TST01TP,@1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4506  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                          MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4509  06                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450A  01                            .BYTE   @1
 (1)  450B  CD  15  28      18.0          TST01TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  450E  DA  7F  43      10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1818                                ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1819  4511  7A                 4.0    TST01SZ: MOV    A,D               ;GET THE CHARACTER COUNT
1820  4512  3C                 4.0          INR     A                   ;INCREMENT
1821  4513  FE  08              7.0          CPI     $8                  ;DONE?
1822  4515  CA  1C  45      10.0          JZ      TST01N                  ;YES-GO CHECK ECC
1823  4518  57                 4.0          MOV     D,A                 ;NO-CONTINUE
1824  4519  C3  B8  44      10.0          JMP     TST01Y
1825
1826  451C  D3  0C            10.0    TST01N: OUT     RINST           ;SINGLE STEP THE READ PATH
1827  451E  3A  E2  48        13.0            LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1828  4521                                    ROUT    EDATA           ;SAVE IN CAS
 (1)  4521  D3  95            10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4523  7F                 4.0                    MOV     A,A             ;RETRY LINK
1829  4524  47                 4.0            MOV     B,A             ;SAVE IN REGISTER B
1830  4525  DB  19            10.0            IN      ECCCOR
1831  4527                                    ROUT    ADATA
 (1)  4527  D3  94            10.0                    OU1     ADATA           ;WRITE AC INTO ADATA
 (1)  4529  7F                 4.0                    MOV     A,A             ;RETRY LINK
1832  452A  B8                 4.0            CMP     B
1833  452B  CA  33  45        10.0            JZ      TST01U
1834  452E                                    ERR     TST01C,TST01U,1
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  452E  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                            MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4531  07                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4532  01                              .BYTE   1
 (1)  4533  CD  15  28        18.0          TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4536  DA  7F  43        10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1835                                ;>CORRECTED ECC CHARACTER INCORRECT
1836  4539  DB  1A            10.0            IN      ECCSTA          ;GET THE ECC STATUS
1837  453B  E6  0F             7.0            ANI     @17             ;REMOVE UNWANTED BITS
1838  453D  FE  02             7.0            CPI     E.TTEC          ;IS TWO TRACK CORRECT BIT SET?
```

```
1839  453F  CA  4F  45   10.0          JZ      TST01V
1840  4542                             ROUT    ADATA
 (1)  4542  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4544  7F             4.0                 MOV     A,A             ;RETRY LINK
1841  4545  3E  02         7.0          MVI     A,E.TTEC
1842  4547                             ROUT    EDATA
 (1)  4547  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4549  7F             4.0                 MOV     A,A             ;RETRY LINK
1843  454A                             ERRB    TST01C,TST01V,1
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  454A  CD  12  28    18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454D  08                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454E  01                                .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  454F  CD  15  28    18.0          TST01V.:        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4552  DA  7F  43    10.0                 JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1844                             ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1845  4555  3A  54  49    13.0          LDA     INDAT
1846  4558  3C                 4.0          INR     A
1847  4559  32  54  49    13.0          STA     INDAT
1848  455C  C2  7F  43    10.0          JNZ     TST01C
1849  455F                             ENDTST  TST01X
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  455F                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  455F  CD  06  28    18.0                 CALL    REQST
 (2)  4562  00                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4563  00  00                            .WORD                   ;SYSTEM '' COUNT
 (2)  4565  00  00                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4567  00                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4568  07                                .BYTE   7               ;REQUEST CODE
 (1)  4569  3A  9A  4F    13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  456C  3D             4.0          DCR     A               ;DOWNCOUNT
 (1)  456D  32  9A  4F    13.0          STA     ITERA           ;SAVE COUNT
 (1)  4570  F2  51  43    10.0          JP      TST01X          ;DO TEST UNTIL TILL = 0
```

```
1852
1853  4573                    .SBTTL  TEST 2 - ECC BAD DATA TRACKS 1 AND 9 NINE SOFT POINTERS
 (1)                        ST
 (1)                        ;*********************************************************************
 (1)                        ;*TEST TITLE
1854                        ;*---- -----
1855  4573                  ;*ECC BAD DATA TRACKS 1 AND 9 NINE SOFT POINTERS
 (1)                        SD
 (1)                        ;*********************************************************************
 (1)                        ;*DESCRIPTION
1856                        ;*-----------
1857                        ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO GIVE PRIORITY
1858                        ;*FOR DATA CORRECTION TO THE EDGE TRACKS (1 AND 9) WHEN GIVEN A MULTITUDE
1859                        ;*OF POINTERS WITH EQUAL VALUE.  THE PROGRAM FORCES BAD DATA ON TRACKS
1860                        ;*1 AND 9 (DATA BIT POSITIONS 2 AND 3) AND PRESENTS THE ECC WITH 9
1861                        ;*POINTERS OF EQUAL VALUE (SEE TABLE).  THE PROGRAM CHECKS FOR CORRECTED
1862                        ;*DATA AND THE TWO TRACK ERROR CORRECT STATUS BIT FROM THE ECC.
1863                        ;*
1864                        ;*      VALUE OF THE                    POINTER GENERATED
1865                        ;*       BYTE COUNT                     ----------------
1866                        ;*      ------------
1867                        ;*
1868                        ;*        XXXXXX00                          HISTORY
1869                        ;*        XXXXXX01                          PHASE TIE
1870                        ;*        XXXXXX10                          AMTIE
1871                        ;*        XXXXXX11                          HISTORY
1872                        ;*
1873                        ;*      X = DON'T CARE
1874                        ;*
1875                        ;*      BYTE COUNT                              DATA
1876                        ;*      ----------                              ----
1877                        ;*
1878                        ;*          001                   000,001,002,003,004,005,006
1879                        ;*          002                   007,010,011,012,013,014,015
1880                        ;*          003                   016,017,020,021,022,023,024
1881                        ;*          004                   025,026,027,030,031,032,033
1882                        ;*          005                   034,035,036,037,040,041,042
1883                        ;*          006                   043,044,045,046,047,050,051
1884                        ;*          007                   052,053,054,055,056,057,060
1885                        ;*          010                   061,062,063,064,065,066,067
1886                        ;*          011                   070,071,072,073,074,075,076
1887                        ;*          012                   077,100,101,102,103,104,105
1888                        ;*          013                   106,107,110,111,112,113,114
1889                        ;*          014                   115,116,117,120,121,122,123
1890                        ;*          015                   124,125,126,127,130,131,132
1891                        ;*          016                   133,134,135,136,137,140,141
1892                        ;*          017                   142,143,144,145,146,147,150
1893                        ;*          020                   151,152,153,154,155,156,157
1894                        ;*          021                   160,161,162,163,164,165,166
1895                        ;*          022                   167,170,171,172,173,174,175
1896                        ;*          023                   176,177,200,201,202,203,204
1897                        ;*          024                   205,206,207,210,211,212,213
1898                        ;*          025                   214,215,216,217,220,221,222
1899                        ;*          026                   223,224,225,226,227,230,231
                            ;*          027                   232,233,234,235,236,237,240
```

```
                                ;*          030                 241,242,243,244,245,246,247
                                ;*          031                 250,251,252,253,254,255,256
                                ;*          032                 257,260,261,262,263,264,265
                                ;*          033                 266,267,270,271,272,273,274
                                ;*          034                 275,276,277,300,301,302,303
                                ;*          035                 304,305,306,307,310,311,312
                                ;*          036                 313,314,315,316,317,320,321
                                ;*          037                 322,323,324,325,326,327,330
                                ;*          040                 331,332,333,334,335,336,337
                                ;*          041                 340,341,342,343,344,345,346
                                ;*          042                 347,350,351,352,353,354,355
                                ;*          043                 356,357,360,361,362,363,364
                                ;*          044                 365,366,367,370,371,372,373
                                ;*          045                 374,375,376,377,000,001,002
 4573                           SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*    DELAY 50 MICROSECONDS
                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*    CLEAR THE INPUT DATA
                                ;*    CLEAR BAD TRANSLATION POINTERS
                                ;*  BGNDO
                                ;*  : ENABLE THE READ PATH CLOCK
                                ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  : CLOCK THE FIFO'S
                                ;*  : ISSUE CLEAR ALL COMMAND
                                ;*  : WAIT
                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                                ;*  : CALL SUBROUTINE CLECC
                                ;*  : SET THE LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  GET THE INPUT DATA BYTE
                                ;*  : :  STORE IN BUFFER
                                ;*  : :  CALL SUBROUTINE ECC
                                ;*  : :  DECREMENT THE LOOP COUNT
                                ;*  : :  INCREMENT THE INPUT DATA
                                ;*  : :  DO UNTIL THE LOOP COUNT=0
                                ;*  : ENDDO
                                ;*  : SET THE LOOP COUNT TO 3
                                ;*  : BGNDO
                                ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER FROM THE DATA BUFFER
                                ;*  : :  CALL SUBROUTINE T4X5
                                ;*  : :  INIT THE LOOP COUNT TO 5
                                ;*  : :  BGNDO
                                ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
```

```
1951                                    ;*  : : :  DECREMENT THE LOOP COUNT
1952                                    ;*  : : :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1953                                    ;*  : : :  DO UNTIL LOOP COUNT = 0
1954                                    ;*  : : ENDDO
1955                                    ;*  : : FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER FROM THE DATA BUFFER
1956                                    ;*  : : FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1957                                    ;*  : : CALL SUBROUTINE T4X5
1958                                    ;*  : : INIT THE LOOP COUNT TO 5
1959                                    ;*  : : BGNDO
1960                                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1961                                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1962                                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1963                                    ;*  : : :  DECREMENT THE LOOP COUNT
1964                                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1965                                    ;*  : : ENDDO
1966                                    ;*  : : DECREMENT THE LOOP COUNT
1967                                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1968                                    ;*  : ENDDO
1969                                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1970                                    ;*  : SET UP WATCHDOG TIMER COUNT
1971                                    ;*  : BGNDO
1972                                    ;*  : :    SINGLE STEP THE READ PATH
1973                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1974                                    ;*  : :    DO UNTIL WATCHDOG TIMER - 0 OR DATA READY SETS
1975                                    ;*  : ENDDO
1976                                    ;*  : IF WATCHDOG TIMER=0
1977                                    ;*  : :    THEN-ERROR
1978                                    ;*  : :    ELSE-CONTINUE
1979                                    ;*  : ENDIF
1980                                    ;*  : GET THE TIE BUS TABLE ADDRESS
1981                                    ;*  : SINGLE STEP THE READ PATH
1982                                    ;*  : INIT THE LOOP COUNT TO 0
1983                                    ;*  : BGNDO
1984                                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1, 2 OR 3
1985                                    ;*  : :    :  THEN-LOAD TIE BUS WITH 10(8)
1986                                    ;*  : :    :  ELSE-CLEAR TIE BUS POINTER
1987                                    ;*  : :    ENDIF
1988                                    ;*  : :    SINGLE STEP THE READ PATH
1989                                    ;*  : :    DECREMENT THE LOOP COUNT
1990                                    ;*  : :    DO UNTIL LOOP COUNT = 9
1991                                    ;*  : ENDDO
1992                                    ;*  : SET UP WATCHDOG TIMER COUNT
1993                                    ;*  : BGNDO
1994                                    ;*  : :  SINGLE STEP THE READ PATH
1995                                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1996                                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1997                                    ;*  : ENDDO
1998                                    ;*  : IF WATCHDOG TIMER=0
1999                                    ;*  : :  THEN-ERROR-EXIT TEST
2000                                    ;*  : :  ELSE-CONTINUE
2001                                    ;*  : ENDIF
2002                                    ;*  : SINGLE STEP THE READ PATH
2003                                    ;*  : INIT THE  LOOP COUNT TO 7
2004                                    ;*  : BGNDO
```

```
2005                                    ;*  : :  SINGLE STEP THE READ PATH
2006                                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
2007                                    ;*  : :  IF NOT EQUAL
2008                                    ;*  : :  :  THEN-ECC ERROR
2009                                    ;*  : :  :  ELSE-CONTINUE
2010                                    ;*  : :  ENDIF
2011                                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
2012                                    ;*  : :  IF NOT EQUAL
2013                                    ;*  : :  :  THEN-ECC ERROR
2014                                    ;*  : :  :  ELSE-CONTINUE
2015                                    ;*  : :  ENDIF
2016                                    ;*  : :  DECREMENT THE LOOP COUNT
2017                                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
2018                                    ;*  : ENDDO
2019                                    ;*  : SINGLE STEP THE READ PATH
2020                                    ;*  : IF EXPECTED NOT EQUAL
2021                                    ;*  : :  THEN-ECC ERROR
2022                                    ;*  : :  ELSE-CONTINUE
2023                                    ;*  : ENDIF
2024                                    ;*  : INPUT THE ECC STATUS
2025                                    ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
2026                                    ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
2027                                    ;*  : :  THEN-CONTINUE
2028                                    ;*  : :  ELSE-ERROR
2029                                    ;*  : ENDIF
2030                                    ;*  : INCREMENT THE INPUT DATA
2031                                    ;*  : DO UNTIL INPUT DATA = 0
2032                                    ;*  ENDDO
2033                                    ;*ENDTST
2034  4573                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2035                                    ;*ECC5 MICRO TEST 02
2036                                    ;*ECC5 MICRO ERROR 11
2037                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2038                                    ;*M8951, M8950'S
2039                                    ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
2040                                    ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2041                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2042                                    ;*FATAL ERROR - MICRO TEST ABORTED
2043                                    ;*
2044                                    ;*ECC5 MICRO TEST 02
2045                                    ;*ECC5 MICRO ERROR 12
2046                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2047                                    ;*M8951, M8950'S
2048                                    ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2049                                    ;*SENT TO THE ECC
2050                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2051                                    ;*FATAL ERROR - MICRO TEST ABORTED
2052                                    ;*
2053                                    ;*ECC5 MICRO TEST 02
2054                                    ;*ECC5 MICRO ERROR 13
2055                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
```

```
2056                                    ;*M8951, M8950'S
2057                                    ;*DATA FROM THE ECC-INCORRECT
2058                                    ;*DATA SHOULD NOT BE CHANGED
2059                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2060                                    ;*ACTUAL = NNNN
2061                                    ;*EXPECTED = NNNN
2062                                    ;*
2063                                    ;*ECC5 MICRO TEST 02
2064                                    ;*ECC5 MICRO ERROR 14
2065                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2066                                    ;*M8951, M8950'S
2067                                    ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2068                                    ;*PARITY SHOULD NOT BE CHANGED
2069                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2070                                    ;*
2071                                    ;*ECC5 MICRO TEST 02
2072                                    ;*ECC5 MICRO ERROR 15
2073                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2074                                    ;*M8951, M8950'S
2075                                    ;*CORRECTED DATA PARITY BIT = ONE SHOULD - ZERO
2076                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2077                                    ;*
2078                                    ;*ECC5 MICRO TEST 02
2079                                    ;*ECC5 MICRO ERROR 16
2080                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2081                                    ;*M8951, M8950'S
2082                                    ;*CORRECTED ECC CHARACTER INCORRECT
2083                                    ;*ECC CHARACTER SHOULD NOT BE CHANGED
2084                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2085                                    ;*
2086                                    ;*ECC5 MICRO TEST 02
2087                                    ;*ECC5 MICRO ERROR 17
2088                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2089                                    ;*M8951, M8950'S
2090                                    ;*ECC STATUS INCORRECT AFTER A DATA GROUP
2091                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2092                                    ;*ACTUAL = NNNN
2093                                    ;*EXPECTED = NNNN
2094  4573                              S
 (1)                                    ; ***************************************************************************************
2095
2096  4573                              TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  4573  3E  02          7.0                  MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4575  CD  03  28      18.0                 CALL    TEST            ;SETUP THE TEST
2097                                    ;%ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2098                                    ;&M8951, M8950'S
2099
2100  4578  3A  53  49      13.0          LDA     SUNIT
2101  457B  D3  E0          10.0          OUT     MBSEL
2102  457D  3E  10           7.0  TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
2103  457F  D3  F0          10.0          OUT     CLKCTL
2104  4581  3E  10           7.0          MVI     A,W.GCR
2105  4583  D3  D3          10.0          OUT     WMCCTL
2106  4585  3E  04           7.0          MVI     A,P.RPEN
```

```
2107  4587  D3  4C          10.0          OUT     PENAB
2108
2109  4589  3E  60           7.0          MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
2110  458B  D3  48          10.0          OUT     PDIAG           ;LCS, LWR
2111  458D  AF                4.0          XRA     A
2112  458E  D3  44          10.0          OUT     TAMT
2113  4590  3E  0A           7.0          MVI     A,10            ;DELAY
2114  4592  3D                4.0  1$:     DCR     A               ;50
2115  4593  C2  92  45      10.0          JNZ     1$              ;MICROSECONDS
2116  4596  AF                4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2117  4597  D3  D2          10.0          OUT     TRKENA          ;CLEAR ALL TRACKS FROM
2118  4599  D3  0A          10.0          OUT     RTIEB           ;CLEAR THE TIE BUS REGISTER
2119  459B  D3  D2          10.0          OUT     TRKENA
2120  459D  32  54  49      13.0          STA     INDAT           ;CLEAR THE INPUT DATA BYTE
2121  45A0                                ROUT    R05H
 (1)  45A0  D3  8B          10.0                  OUT     R05H            ;WRITE AC INTO R05H
 (1)  45A2  7F                4.0                  MOV     A,A             ;RETRY LINK
2122  45A3  32  B3  48      13.0          STA     BAD3            ;NO THIRD FAILURE
2123  45A6  3E  03           7.0          MVI     A,3
2124  45A8  32  B1  48      13.0          STA     BAD1            ;DATA BIT 2 BAD
2125  45AB  3E  04           7.0          MVI     A,4
2126  45AD  32  B2  48      13.0          STA     BAD2            ;DATA BIT 3 BAD
2127  45B0  3E  A8           7.0  TST02C: MVI     A,R.PLO1!R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2128                                                               ;SET TIE BUS JAM AND PLO DISABLE
2129  45B2  D3  09          10.0          OUT     RPCTL
2130  45B4  D3  08          10.0          OUT     RFIFOL          ;CLOCK THE FIFO'S
2131  45B6  3E  0D           7.0          MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
2132  45B8  D3  0B          10.0          OUT     RCMD
2133
2134  45BA  00                4.0          NOP                     ;WAIT
2135  45BB  00                4.0          NOP
2136  45BC  00                4.0          NOP
2137  45BD  00                4.0          NOP
2138  45BE  00                4.0          NOP
2139
2140  45BF  3E  A9           7.0          MVI     A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2141                                                               ;SET TIE BUS JAM AND PLO DISABLE
2142  45C1  D3  09          10.0          OUT     RPCTL
2143  45C3  CD  B4  48      18.0          CALL    CLECC           ;CLEAR THE ECC CHARACTER
2144  45C6  21  5D  49      10.0          LXI     H,BUFFER        ;SET UP THE BUFFER POINTER
2145  45C9  16  07           7.0          MVI     D,@7            ;SET UP THE LOOP COUNT
2146  45CB  3A  54  49      13.0  TST02F: LDA     INDAT           ;GET THE INPUT DATA
2147  45CE  77                7.0          MOV     M,A             ;SAVE CHARACTER IN THE BUFFER
2148  45CF  3C                4.0          INR     A               ;UPDATE THE DATA
2149  45D0  32  54  49      13.0          STA     INDAT           ;SAVE IT
2150  45D3  23                6.0          INX     H               ;UPDATE THE BUFFER POINTER
2151  45D4  3D                4.0          DCR     A               ;RECREATED THE DATA FOR THE ECC SUBROUTINE
2152  45D5  CD  B9  48      18.0          CALL    ECC             ;CALCULATE ECC FOR
2153  45D8  15                4.0          DCR     D               ;THE 7 INPUT BYTES
2154  45D9  C2  CB  45      10.0          JNZ     TST02F
2155
2156  45DC  0E  03           7.0          MVI     C,@3            ;INIT THE LOOP COUNT
2157  45DE  3A  54  49      13.0  TST02I: LDA     INDAT           ;GET THE INPUT DATA
2158  45E1  06  25           7.0          MVI     B,@45
```

```
2159  45E3  FE 03        7.0          CPI   @3
2160  45E5  CA F1 45     10.0         JZ    TST02Z
2161  45E8  06 00        7.0          MVI   B,0
2162  45EA  04           4.0  TST02W: INR   B
2163  45EB  D6 07        7.0          SUI   @7
2164  45ED  A7           4.0          ANA   A
2165  45EE  C2 EA 45     10.0         JNZ   TST02W
2166  45F1  78           4.0  TST02Z: MOV   A,B
2167  45F2                            ROUT  R05L
  (1) 45F2  D3 8A        10.0                OUT   R05L          ;WRITE AC INTO R05L
  (1) 45F4  7F           4.0                 MOV   A,A           ;RETRY LINK
2168  45F5  32 4F 49     13.0         STA   TEMPX
2169  45F8  3A 5D 49     13.0         LDA   BUFFER
2170  45FB  32 A1 48     13.0         STA   TRNIN          ;FILL THE TRANSLATOR
2171  45FE  3A 5E 49     13.0         LDA   BUFFER+1
2172  4601  32 A2 48     13.0         STA   TRNIN+1        ;SUBROUTINE INPUT
2173  4604  3A 5F 49     13.0         LDA   BUFFER+2
2174  4607  32 A3 48     13.0         STA   TRNIN+2        ;BUFFER
2175  460A  3A 60 49     13.0         LDA   BUFFER+3
2176  460D  32 A4 48     13.0         STA   TRNIN+3        ;
2177  4610  CD CE 47     18.0         CALL  BADT4X5        ;TRANSLATE THE SUBGROUP
2178
2179  4613  21 A5 48     10.0         LXI   H,TRNOUT       ;GET POINTER TO TRANSLATED DATA TABLE
2180  4616  06 05        7.0          MVI   B,@5           ;SET UP LOOP COUNT
2181  4618  7E           7.0  TST02G: MOV   A,M           ;GET A DATA BYTE
2182  4619  D3 40        10.0         OUT   TCMD           ;STORE DATA IN COMMAND ADDRESS
2183  461B  23           6.0          INX   H              ;POINT TO DATA PARITY
2184  461C  7E           7.0          MOV   A,M            ;GET THE DATA PARITY
2185
2186  461D  07           4.0          RLC                  ;POSITION FOR OUTPUT
2187  461E  F6 60        7.0          ORI   P.LWR.P.LCS    ;OR IN CONTROL BITS
2188  4620  D3 48        10.0         OUT   PDIAG          ;OUTPUT THE DATA PARITY
2189
2190  4622  D3 08        10.0         OUT   RFIFOL         ;CLOCK DATA INTO THE FIFO'S
2191
2192  4624  23           6.0          INX   H              ;UPDATE THE TABLE POINTER
2193  4625  05           4.0          DCR   B              ;DECREMENT LOOP COUNT
2194  4626  C2 18 46     10.0         JNZ   TST02G         ;DO UNTIL LOOP COUNT = 0
2195
2196  4629  3A 61 49     13.0         LDA   BUFFER+4
2197  462C  32 A1 48     13.0         STA   TRNIN          ;FILL THE FIRST THREE
2198  462F  3A 62 49     13.0         LDA   BUFFER+5
2199  4632  32 A2 48     13.0         STA   TRNIN+1        ;BYTES OF THE TRANSLATOR
2200  4635  3A 63 49     13.0         LDA   BUFFER+6
2201  4638  32 A3 48     13.0         STA   TRNIN+2        ;SUBROUTINE WITH THE INPUT DATA
2202  463B  3A E2 48     13.0         LDA   ECCCHR         ;STORE THE CALCULATED
2203  463E  32 A4 48     13.0         STA   TRNIN+3        ;ECC CHARACTER AS THE LAST CHARACTER
2204
2205  4641  CD CE 47     18.0         CALL  BADT4X5        ;TRANSLATE THE SECOND SUBGROUP
2206
2207  4644  21 A5 48     10.0         LXI   H,TRNOUT       ;GET THE POINTER TO INPUT DATA TABLE
2208  4647  06 05        7.0          MVI   B,@5           ;SET UP THE LOOP COUNT
2209  4649  7E           7.0  TST02H: MOV   A,M           ;GET A DATA BYTE
2210  464A  D3 40        10.0         OUT   TCMD           ;STORE DATA IN CMD ADDRESS
```

```
2211  464C  23          6.0          INX   H                ;POINT TO DATA PARITY BIT
2212  464D  7E          7.0          MOV   A,M              ;GET THE DATA PARITY
2213  464E  07          4.0          RLC                    ;POSITION FOR OUTPUT
2214  464F  F6  60      7.0          ORI   P.LWR!P.LCS      ;OR IN CONTROL BITS
2215  4651  D3  48     10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
2216  4653  D3  08     10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFOS
2217  4655  23          6.0          INX   H                ;UPDATE THE TABLE POINTER
2218  4656  05          4.0          DCR   B                ;DECREMENT THE LOOP COUNT
2219  4657  C2  49  46  10.0          JNZ   TST02H           ;DO UNTIL LOOP COUNT=0
2220  465A  0D          4.0          DCR   C                ;PUT THE GROUP IN
2221  465B  C2  DE  45  10.0          JNZ   TST02I           ;THE FIFO TWICE
2222
2223  465E  3E  0B       7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
2224  4660  D3  0B      10.0          OUT   RCMD             ;COMMAND
2225  4662  11  01  00  10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
2226  4665  21  A8  FD  10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
2227  4668  D3  0C      10.0  2$:     OUT   RINST            ;STEP THE READ PATH
2228  466A  DB  01      10.0          IN    RPCHI            ;DATA READY SET?
2229  466C  E6  10       7.0          ANI   R.DRDY
2230  466E  C2  83  46  10.0          JNZ   TST02A           ;YES-GO PROCESS
2231  4671  19          10.0          DAD   D                ;WATCH DOG TIMEOUT?
2232  4672  D2  68  46  10.0          JNC   2$               ;NO-CONTINUE
2233  4675                         ERR   RETRY,DUMMU,1
  (1)                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4675  CD  09  28  18.0                 CALL  ERLP           ;PROCESS ERROR - DO 2.3
  (1)        0009                             MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4678  09                               .BYTE MSGN           ;MESSAGE NUMBER ID
  (1)  4679  01                               .BYTE 1
  (1)  467A  CD  15  28  18.0       DUMMU:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
  (1)  467D  DA  98  47  10.0               JC    RETRY          ;LOOP ADDRESS IF LOOP SPECIFIED
2234                           ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2235                           ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2236                           ;<FATAL ERROR - MICRO TEST ABORTED
2237  4680  C3  95  47  10.0          JMP   EXIT
2238
2239  4683  D3  0C      10.0  TST02A: OUT   RINST
2240  4685  06  00       7.0          MVI   B,$0             ;INIT THE BIT TIME COUNT (LOOP COUNT)
2241  4687  00          4.0  TST02B: NOP
2242  4688  CD  79  48  18.0  6$:     CALL  POINTER
2243  468B  D3  0C      10.0          OUT   RINST            ;STEP CLOCK
2244  468D  04          4.0          INR   B                ;DECREMENT THE LOOP COUNT
2245  468E  78          4.0          MOV   A,B
2246  468F  FE  09       7.0          CPI   $9               ;DONE
2247  4691  C2  87  46  10.0          JNZ   TST02B           ;DO UNTIL LOOP COUNT = ZERO
2248  4694  AF          4.0          XRA   A
2249  4695  D3  0A      10.0          OUT   RTIEB            ;CLEAR THE TIE BUS REGISTER
2250  4697  D3  0C      10.0          OUT   RINST            ;CLOCK THE READ PATH
2251  4699  D3  0C      10.0          OUT   RINST            ;CLOCK THE READ PATH
2252  469B  11  01  00  10.0          LXI   D,1              ;SET WATCHDOG TIMER INCREMENT
2253  469E  21  A8  FD  10.0          LXI   H,-600           ;SET WATCHDOG COUNT TO 600
2254  46A1  D3  0C      10.0  3$:     OUT   RINST            ;SINGLE STEP THE READ PATH
2255  46A3  DB  01      10.0          IN    RPCHI            ;DATA READY SET?
2256  46A5  E6  10       7.0          ANI   R.DRDY
2257  46A7  C2  BC  46  10.0          JNZ   TST02K           ;YES-GO PROCESS
```

```
2258  46AA  19              10.0          DAD     D                       ;NO-WATCHDOG TIMEOUT?
2259  46AB  D2  A1  46      10.0          JNC     3$                      ;NO-CONTINUE
2260  46AE                                ERR     RETRY,DUMMV,1
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46AE  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                        MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46B1  0A                          .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  46B2  01                          .BYTE   1
 (1)  46B3  CD  15  28      18.0          DUMMV:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46B6  DA  98  47      10.0                  JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2261                                ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2262                                ;>SENT TO THE ECC
2263                                ;<FATAL ERROR - MICRO TEST ABORTED
2264  46B9  C3  95  47      10.0          JMP     EXIT
2265  46BC  16  01           7.0    TST02K: MVI     D,@1                    ;INIT THE LOOP COUNT
2266  46BE  21  5D  49       10.0          LXI     H,BUFFER
2267  46C1  AF                4.0    TST02Y: XRA     A
2268  46C2                                ROUT    R05H
 (1)  46C2  D3  88           10.0                  OUT     R05H                    ;WRITE AC INTO R05H
 (1)  46C4  7F               4.0                   MOV     A,A                     ;RETRY LINK
2269  46C5  7E               7.0           MOV     A,M                     ;GET THE INPUT DATA
2270  46C6                                ROUT    EDATA
 (1)  46C6  D3  95           10.0                  OUT     EDATA                   ;WRITE AC INTO EDATA
 (1)  46C8  7F               4.0                   MOV     A,A                     ;RETRY LINK
2271  46C9  A7               4.0           ANA     A                       ;SET CONDITION BITS
2272  46CA  3E  00            7.0           MVI     A,0                     ;CLEAR THE ACC.
2273  46CC  E2  D0  46       10.0          JPO     1$                      ;ODD # OF ONES - CONTINUE
2274  46CF  3C               4.0           INR     A                       ;EVEN # OF ONES - SET PARITY
2275  46D0  32  55  49       13.0    1$:   STA     INDATP                  ;STORE THE PARITY BIT
2276  46D3  D3  0C           10.0    TST02L: OUT     RINST                   ;SINGLE STEP THE READ LOGIC
2277  46D5  7E               7.0           MOV     A,M
2278  46D6  23               6.0           INX     H
2279  46D7  47               4.0           MOV     B,A
2280  46D8  DB  19           10.0          IN      ECCCOR
2281  46DA  B8               4.0           CMP     B
2282  46DB  CA  EA  46       10.0          JZ      TST02M
2283  46DE                                ROUT    ADATA
 (1)  46DE  D3  94           10.0                  OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  46E0  7F               4.0                   MOV     A,A                     ;RETRY LINK
2284  46E1  78               4.0           MOV     A,B                     ;GET THE BYTE COUNT
2285  46E2                                ROUT    R05L                    ;STORE IN CAS
 (1)  46E2  D3  8A           10.0                  OUT     R05L                    ;WRITE AC INTO R05L
 (1)  46E4  7F               4.0                   MOV     A,A                     ;RETRY LINK
2286  46E5                                ERRB    RETRY,TST02M,@1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46E5  CD  12  28      18.0                  CALL    ERLPB                   ;PROCESS ERROR - DO 2.3
 (1)        000B                        MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E8  0B                          .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  46E9  01                          .BYTE   @1                      ;PRINT ROUTINE NUMBER
 (1)  46EA  CD  15  28      18.0                TST02M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46ED  DA  98  47      10.0                  JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2287                                ;>DATA FROM THE ECC-INCORRECT
2288                                ;>DATA SHOULD NOT BE CHANGED
2289  46F0  DB  15           10.0          IN      RPSTA                   ;GET THE ECC PARITY BIT
```

```
2290  46F2  E6  80        7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2291  46F4  07            4.0          RLC                     ;POSITION BIT FOR COMPARE
2292  46F5  4F            4.0          MOV     C,A             ;SAVE IN REG. C
2293  46F6  3A  55  49     13.0         LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2294  46F9  A7            4.0          ANA     A               ;SET THE CONDITION BITS
2295  46FA  CA  15  47     10.0         JZ      TST02QO         ;GO CHECK FOR PARITY=0
2296  46FD  79            4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
2297  46FE  FE  01        7.0          CPI     @1
2298  4700  CA  0C  47     10.0         JZ      TST02RG
2299  4703  78            4.0          MOV     A,B
2300  4704                             ROUT    R05L
 (1)  4704  D3  8A        10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  4706  7F            4.0                  MOV     A,A             ;RETRY LINK
2301  4707                             ERR     RETRY,TST02RG,@1
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4707  CD  09  28     18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  470A  0C                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4708  01                                 .BYTE   @1
 (1)  470C  CD  15  28     18.0         TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  470F  DA  98  47     10.0                 JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2302                             ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2303                             ;>PARITY SHOULD NOT BE CHANGED
2304  4712  C3  29  47     10.0         JMP     TST02SZ         ;CONTINUE WITH TEST
2305  4715  79            4.0  TST02QO: MOV     A,C             ;CHECK FOR PARITY = ZERO
2306  4716  A7            4.0          ANA     A               ;SET CONDITION BITS
2307  4717  CA  23  47     10.0         JZ      TST02TP         ;CONTINUE IF ZERO
2308  471A  78            4.0          MOV     A,B
2309  471B                             ROUT    R05L
 (1)  471B  D3  8A        10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  471D  7F            4.0                  MOV     A,A             ;RETRY LINK
2310  471E                             ERR     RETRY,TST02TP,@1
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  471E  CD  09  28     18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000D                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4721  0D                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4722  01                                 .BYTE   @1
 (1)  4723  CD  15  28     18.0         TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4726  DA  98  47     10.0                 JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2311                             ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2312  4729  7A            4.0  TST02SZ: MOV     A,D             ;GET THE CHARACTER COUNT
2313  472A  3C            4.0          INR     A               ;INCREMENT
2314  472B  FE  08        7.0          CPI     $8              ;DONE?
2315  472D  CA  34  47     10.0         JZ      TST02N          ;YES-GO CHECK ECC
2316  4730  57            4.0          MOV     D,A             ;NO-CONTINUE
2317  4731  C3  C1  46     10.0         JMP     TST02Y
2318
2319  4734  D3  0C        10.0  TST02N: OUT     RINST           ;SINGLE STEP THE READ PATH
2320  4736  3A  E2  48     13.0         LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
2321  4739                             ROUT    EDATA           ;SAVE IN CAS
 (1)  4739  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  473B  7F            4.0                  MOV     A,A             ;RETRY LINK
2322  473C  47            4.0          MOV     B,A             ;SAVE IN REGISTER B
2323  473D  DB  19        10.0         IN      ECCCOR
```

```
2324  473F                                          ROUT    ADATA
 (1)  473F  D3  94              10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4741  7F                   4.0                       MOV     A,A             ;RETRY LINK
2325  4742  B8                   4.0                CMP     B
2326  4743  CA  51  47           10.0                JZ      TST02U
2327  4746  3A  54  49           13.0                LDA     INDAT           ;GET THE DATA/COUNT
2328  4749                                          ROUT    R05L            ;SAVE IN THE CAS
 (1)  4749  D3  8A              10.0                       OUT     R05L            ;WRITE AC INTO R05L
 (1)  474B  7F                   4.0                       MOV     A,A             ;RETRY LINK
2329  474C                                          ERR     RETRY,TST02U,1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  474C  CD  09  28          18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000E                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  474F  0E                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4750  01                                           .BYTE   1
 (1)  4751  CD  15  28          18.0                TST02U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4754  DA  98  47          10.0                       JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2330                                          ;>CORRECTED ECC CHARACTER INCORRECT
2331                                          ;>ECC CHARACTER SHOULD NOT BE CHANGED
2332  4757  DB  1A                10.0                IN      ECCSTA          ;GET THE ECC STATUS
2333  4759  E6  0F                 7.0                ANI     @17             ;REMOVE UNWANTED BITS
2334  475B  FE  02                 7.0                CPI     E.TTEC          ;IS UNCORRECTABLE BIT SET?
2335  475D  CA  73  47             10.0                JZ      TST02V
2336  4760                                          ROUT    ADATA
 (1)  4760  D3  94              10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4762  7F                   4.0                       MOV     A,A             ;RETRY LINK
2337  4763  3E  02                 7.0                MVI     A,E.TTEC
2338  4765                                          ROUT    EDATA
 (1)  4765  D3  95              10.0                       OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4767  7F                   4.0                       MOV     A,A             ;RETRY LINK
2339  4768  3A  54  49           13.0                LDA     INDAT           ;GET THE DATA/COUNT
2340  476B                                          ROUT    R05L            ;SAVE IN THE CAS
 (1)  476B  D3  8A              10.0                       OUT     R05L            ;WRITE AC INTO R05L
 (1)  476D  7F                   4.0                       MOV     A,A             ;RETRY LINK
2341  476E                                          ERRB    RETRY,TST02V,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  476E  CD  12  28          18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4771  0F                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4772  01                                           .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4773  CD  15  28          18.0                TST02V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4776  DA  98  47          10.0                       JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2342                                          ;>ECC STATUS INCORRECT AFTER A DATA GROUP
2343  4779  3A  54  49           13.0                LDA     INDAT
2344  477C  FE  03                 7.0                CPI     @3              ;DONE
2345  477E  C2  B0  45           10.0                JNZ     TST02C
```

```
                                                ENDTST  TST02X
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4781                                                      CALL    REQST
 (2)  4781  CD  06  28      18.0                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4784  00                                                  .WORD                   ;SYSTEM '#' COUNT
 (2)  4785  00  00                                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4787  00  00                                              .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4789  00                                                  .BYTE   7               ;REQUEST CODE
 (2)  478A  07
 (1)  478B  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  478E  3D               4.0                DCR     A               ;DOWNCOUNT
 (1)  478F  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
 (1)  4792  F2  7D  45      10.0                JP      TST02X          ;DO TEST UNTIL TILL - 0
2348
2349  4795  C3  18  28      10.0  EXIT:         JMP     TSTEND
2350  4798  3A  54  49      13.0  RETRY:        LDA     INDAT
2351  479B  D6  07               7.0           SUI     @7
2352  479D  32  54  49      13.0                STA     INDAT
2353  47A0  C3  B0  45      10.0                JMP     TSTC2C
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
 2372  47A3  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
 2373  47A4  C5              12.0          PUSH    B
 2374  47A5  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
 2375  47A7  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
 2376  47A9  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
 2377  47AB  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
 2378  47AC  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
 2379  47AE  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
 2380  47B0  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
 2381  47B2  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
 2382  47B3  D3  40          10.0          OUT     TCMD            ;
 2383  47B5  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
 2384  47B7  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
 2385  47B9  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
 2386  47BB  D3  40          10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
 2387  47BD  AF               4.0          XRA     A
 2388  47BE  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
 2389  47C0  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
 2390  47C2  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
 2391  47C4  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
 2392  47C5  78               4.0          MOV     A,B
 2393  47C6  FE  04           7.0          CPI     4               ;DONE?
 2394  47C8  C2  A7  47      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
 2395  47CB  C1              10.0          POP     B               ;RESET B & C
 2396  47CC  F1              10.0          POP     PSW             ;ALL DONE
 2397  47CD  C9              10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
47CE                            SSUB
                                ;*********************** ************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
47CE                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE, AT THE ENTRY
                                ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                ;*BUT NOT CORRECT FOR THE INPUT DATA SIMULATING BAD DATA.
47CE                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    CLEAR THE BAD TRANSLATION POINTERS
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 1
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    INCREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 10
                                ;*    ENDDO
                                ;*ENDSUB
47CE                            S
                                ;*****************************************************************************

47CE F5              12.0 BADT4X5: PUSH   PSW            ;SAVE REGISTER A + PSW
47CF C3 DD 47        10.0          JMP    BAD
47D2 F5              12.0 T4X5:    PUSH   PSW            ;SAVE REGISTER A + PSW
47D3 AF               4.0          XRA    A              ;CLEAR A
47D4 32 B1 48        13.0          STA    BAD1           ;CLEAR BAD TRANSLATION 1
47D7 32 B2 48        13.0          STA    BAD2           ;CLEAR BAD TRANSLATION 2
47DA 32 B3 48        13.0          STA    BAD3           ;CLEAR BAD TRANSLATION 3
47DD C5              12.0 BAD:     PUSH   B              ;SAVE REGISTER B + C
47DE D5              12.0          PUSH   D              ;SAVE REGISTER D + E
47DF E5              12.0          PUSH   H              ;SAVE REGISTER H + L
```

```
2443                                                ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
2444                                                ;CLEAR THE TRANSLATED DATA TABLE.
2445  47E0  06  0A          7.0           MVI   B,10          ;SET UP LOOP COUNT
2446  47E2  11  A5  48      10.0           LXI   D,TRNOUT      ;GET POINTER TO TRANSLATED DATA TABLE
2447  47E5  AF               4.0           XRA   A             ;CLEAR A
2448  47E6  12               7.0  D4X5:    STAX  D             ;STORE IN THE TABLE
2449  47E7  13               6.0           INX   D             ;UPDATE TABLE POINTER
2450  47E8  05               4.0           DCR   B             ;DECREMENT LOOP COUNT
2451  47E9  C2  E6  47      10.0           JNZ   D4X5          ;DO UNTIL LOOP COUNT=0
2452  47EC  0E  01           7.0           MVI   C,1           ;SET UP TRACK COUNT
2453  47EE  06  04           7.0  B4X5:    MVI   B,4           ;SET UP BIT COUNT
2454  47F0  11  A1  48      10.0           LXI   D,TRNIN
2455  47F3  AF               4.0           XRA   A             ;CLEAR THE GROUP POSITION COUNT
2456  47F4  32  AF  48      13.0           STA   GP4X5
2457  47F7  1A               7.0  C4X5:    LDAX  D             ;GET A DATA BYTE
2458  47F8  A7               4.0           ANA   A             ;SET CONDITION BITS
2459  47F9  E2  FD  47      10.0           JPO   P04X5         ;ODD PARITY LEAVE CARRY CLEAR
2460  47FC  37               4.0           STC                 ;EVEN PARITY SET CARRY
2461  47FD  1F               4.0  P04X5:   RAR                 ;SHIFT OUT DESIRED BIT
2462  47FE  12               7.0           STAX  D             ;STORE RESULT BACK IN TEMP TABLE
2463  47FF  3A  AF  48      13.0           LDA   GP4X5         ;GET THE GROUP POSITION BYTE
2464  4802  17               4.0           RAL                 ;PUT IN THIS DATA BIT
2465  4803  32  AF  48      13.0           STA   GP4X5         ;SAVE THE UPDATED GROUP POSITION BYTE
2466  4806  13               6.0           INX   D             ;UPDATE THE TABLE POINTER
2467  4807  05               4.0           DCR   B             ;DECREMENT THE BIT COUNT
2468  4808  C2  F7  47      10.0           JNZ   C4X5          ;DO UNTIL ALL 4 BITS ARE RECEIVED
2469
2470                                                ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2471
2472                                                ;PERFORM A DOUBLE PRECISION SHIFT RIGHT ON THE TRANSLATED DATA
2473  480B  06  05           7.0           MVI   B,5           ;SET UP LOOP COUNT
2474  480D  11  A5  48      10.0           LXI   D,TRNOUT      ;GET POINTER TO TRANSLATED DATA TABLE
2475  4810  13               6.0  E4X5:    INX   D             ;POINT TO PARITY BIT
2476  4811  1A               7.0           LDAX  D             ;GET PARITY BIT
2477  4812  A7               4.0           ANA   A             ;CLEAR THE CARRY BIT
2478  4813  1F               4.0           RAR                 ;SHIFT TO THE CARRY BIT
2479  4814  12               7.0           STAX  D             ;STORE IT BACK
2480  4815  1B               6.0           DCX   D             ;DECREMENT TO DATA BITS
2481  4816  1A               7.0           LDAX  D             ;GET THE DATA BITS
2482  4817  1F               4.0           RAR                 ;SHIFT IN CARRY BITS
2483  4818  12               7.0           STAX  D             ;STORE BACK IN TABLE
2484  4819  13               6.0           INX   D             ;POINT TO NEXT TABLE ENTRY
2485  481A  13               6.0           INX   D             ;
2486  481B  05               4.0           DCR   B             ;DECREMENT LOOP COUNT
2487  481C  C2  10  48      10.0           JNZ   E4X5          ;DO UNTIL LOOP COUNT=0
2488  481F  3A  B1  48      13.0           LDA   BAD1          ;IS A BAD POINTER SPECIFIED?
2489  4822  A7               4.0           ANA   A             ;SET CONDITION BITS
2490  4823  CA  2A  48      10.0           JZ    BAD1C         ;NO-CONTINUE
2491  4826  B9               4.0           CMP   C             ;YES-IS IT THIS DATA POSITION?
2492  4827  CA  40  48      10.0           JZ    INV4X5        ;YES-GO PERFORM INCORRECT TRANSLATION
2493  482A  3A  B2  48      13.0  BAD1C:   LDA   BAD2          ;IS A BAD POINTER SPECIFIED
2494  482D  A7               4.0           ANA   A             ;SET CONDITION BITS
2495  482E  CA  35  48      10.0           JZ    BAD2C         ;NO-CONTINUE
2496  4831  B9               4.0           CMP   C             ;YES-IS IT THIS DATA POSITION
```

```
2497  4832  CA 40 48   10.0          JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2498  4835  3A B3 48   13.0  BAD2C:  LDA     BAD3            ;IS A BAD POINTER SPECIFIED
2499  4838  A7          4.0          ANA     A               ;SET CONDITION BITS
2500  4839  CA 47 48   10.0          JZ      BAD3C           ;NO-PERFORM PROPER TRANSLATION
2501  483C  B9          4.0          CMP     C               ;YES-IS IT THIS DATA POSITION
2502  483D  C2 47 48   10.0          JNZ     BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
2503  4840  3A AF 48   13.0  INV4X5: LDA     GP4X5           ;GET THE GROUP POSITION COUNT
2504  4843  3C          4.0          INR     A               ;INCREMENT-TO INCORRECT TRANSLATION
2505  4844  32 AF 48   13.0          STA     GP4X5           ;STORE IT
2506  4847  11 91 48   10.0  BAD3C:  LXI     D,.AB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
2507  484A  26 00        7.0          MVI     H,0             ;CLEAR REGISTER H
2508  484C  3A AF 48   13.0          LDA     GP4X5           ;GET GROUP POSITION COUNT
2509  484F  E6 0F        7.0          ANI     $0F
2510  4851  6F          4.0          MOV     L,A
2511  4852  19         10.0          DAD     D               ;ADD GROUP POSITION COUNT TO TABLE
2512  4853  7E          7.0          MOV     A,M             ;GET TRANSLATED DATA
2513  4854  32 B0 48   13.0          STA     TRNTMP
2514                                ;''TRNTMP'' = THE TRANSLATED DATA
2515
2516  4857  11 A5 48   10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2517  485A  06 05        7.0          MVI     B,5             ;SET UP LOOP COUNT
2518  485C  3A B0 48   13.0  F4X5:   LDA     TRNTMP          ;GET TRANSLATED DATA
2519  485F  13          6.0          INX     D               ;POINT TO PARITY POSITION
2520  4860  A7          4.0          ANA     A               ;CLEAR CARRY BIT
2521  4861  1F          4.0          RAR                     ;SHIFT OUT A BIT
2522  4862  32 B0 48   13.0          STA     TRNTMP          ;STORE TRANSLATED DATA
2523  4865  1A          7.0          LDAX    D               ;GET THE PARITY BIT (ALWAYS ZERO)
2524  4866  17          4.0          RAL                     ;ROLL IN THE CARRY BIT
2525  4867  12          7.0          STAX    D               ;STORE AS THE PARITY BIT
2526  4868  13          6.0          INX     D               ;UPDATE TABLE POINTER
2527  4869  05          4.0          DCR     B               ;DECREMENT THE LOOP COUNT
2528  486A  C2 5C 48   10.0          JNZ     F4X5            ;DO UNTIL LOOP COUNT=0
2529
2530                                ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2531
2532  486D  0C          4.0          INR     C               ;DECREMENT THE TRACK COUNT
2533  486E  79          4.0          MOV     A,C
2534  486F  FE 0A        7.0          CPI     10
2535  4871  C2 EE 47   10.0          JNZ     B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2536  4874  E1         10.0          POP     H               ;RESTORE REGISTER H + L
2537  4875  D1         10.0          POP     D               ;RESTORE REGISTER D + E
2538  4876  C1         10.0          POP     B               ;RESTORE REGISTER B + C
2539  4877  F1         10.0          POP     PSW             ;RESTORE REGISTER A + PSW
2540  4878  C9         10.0          RET                     ;RETURN TO USER
2541
```

```
2543                                    .SBTTL  SUBROUTINE CALCULATE POINTER
2544                                    ;*THE POINTERS PROVIDED FOR THE TRACKS ARE ALSO DERIVED FROM THE INPUT/
2545                                    ;*EXPECTED DATA AS PER THE FOLLOWING TABLE.
2546                                    ;*
2547                                    ;*        BYTE COUNT                    POINTER GENERATED
2548                                    ;*      --------------                  -----------------
2549                                    ;*
2550                                    ;*        XXXXXX00                        HISTORY
2551                                    ;*        XXXXXX01                        PHASE TIE
2552                                    ;*        XXXXXX10                        AMTIF
2553                                    ;*        XXXXXX11                        ILLEGAL 5X4
2554                                    ;*
2555                                    ;*      X = DON'T CARE BITS
2556  4879                              SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2557
2558  4879  F5              12.0        POINTER: PUSH   PSW             ;SAVE THE ACCUMULATOR
2559  487A  C5              12.0                 PUSH   B               ;SAVE B + C
2560  487B  3A  4F  49      13.0                 LDA    TEMPX           ;GET BYTE COUNT
2561  487E  E6  03           7.0                 ANI    $3              ;REMOVE THE WORKING BITS
2562  4880  21  8D  48      10.0                 LXI    H,POINTBL       ;GET BASE ADDRESS OF TABLE
2563  4883  4F               4.0                 MOV    C,A
2564  4884  06  00           7.0                 MVI    B,0
2565  4886  09              10.0                 DAD    B
2566  4887  7E               7.0                 MOV    A,M             ;GET THE ECC POINTER
2567  4888  D3  0A          10.0                 OUT    RTIEB           ;SET THE TIE BUS VALUE
2568  488A  C1              10.0                 POP    B               ;RESTORE B + C
2569  488B  F1              10.0                 POP    PSW             ;RESTORE ACCUMULATOR
2570  488C  C9              10.0                 RET
2571
2572  488D  01                           POINTBL: .BYTE  1
2573  488E  02                                    .BYTE  2
2574  488F  04                                    .BYTE  4
2575  4890  0?                                    .BYTE  1
2576
```

```
2578                                .SBTTL  TABLE 4 X 5 TRANSLATION
2579
2580                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2581                                ;  INPUT GROUP POSITIONS          OUTPUT GROUP POSITIONS
2582                                ;   1 2 3 4 / 5 6 7 8              10 9 8 7 6 / 5 4 3 2 1
2583                                ;
2584                                ;       0000                           10011
2585                                ;       0001                           11011
2586                                ;       0010                           01001
2587                                ;       0011                           11001
2588                                ;       0100                           10111
2589                                ;       0101                           10101
2590                                ;       0110                           01101
2591                                ;       0111                           11101
2592                                ;       1000                           01011
2593                                ;       1001                           10010
2594                                ;       1010                           01010
2595                                ;       1011                           11010
2596                                ;       1100                           01111
2597                                ;       1101                           10110
2598                                ;       1110                           01110
2599                                ;       1111                           11110
2600
2601  4891  13                      TAB4X5: .BYTE   @23
2602  4892  1B                              .BYTE   @33
2603  4893  09                              .BYTE   @11
2604  4894  19                              .BYTE   @31
2605  4895  17                              .BYTE   @27
2606  4896  15                              .BYTE   @25
2607  4897  0D                              .BYTE   @15
2608  4898  1D                              .BYTE   @35
2609  4899  0B                              .BYTE   @13
2610  489A  12                              .BYTE   @22
2611  489B  0A                              .BYTE   @12
2612  489C  1A                              .BYTE   @32
2613  489D  0F                              .BYTE   @17
2614  489E  16                              .BYTE   @26
2615  489F  0E                              .BYTE   @16
2616  48A0  1E                              .BYTE   @36
2617
```

```
                                        .SBTTL  SUBROUTINE VARIABLES

                        TRNIN:  .BLKB   4       ;TABLE CONTAINING THE 4 DATA BYTES TO
                                                ;BE TRANSLATED
                        TRNOUT: .BLKB   10      ;TRANSLATED DATA TABLE DATA AFTER 4X5
                                                ;TRANSLATION
                        GP4X5:  .BYTE   0       ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
                                                ;BE TRANSLATED
                        TRNTMP: .BYTE   0       ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
                        BAD1:   .BYTE   0       ;BAD TRANSLATION POINTER 1
                        BAD2:   .BYTE   0       ;BAD TRANSLATION POINTER 2
                        BAD3:   .BYTE   0       ;BAD TRANSLATION POINTER 3
```

```
                                                J 1
                        .SBTTL  SUBROUTINE CLEAR ECC
                        S
                        ; ************************************************************************
                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
                        S
                        ; ************************************************************************

4884  AF                      4.0  CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
4885  32    E2    48        13.0         STA     ECCCHR          ;CLEAR THE ECC CHARACTER
4888  C9                    10.0         RET                     ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        ;
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF ''ECCCHR'' TO UPDATE THE CONTENTS OF ''ECCCHR'' ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        ;
                                        ; ************************************************************************

                                ECC:    PUSH    H               ;SAVE H&L
                                        PUSH    D               ;SAVE D&E
                                        PUSH    PSW             ;SAVE THE ACCUMULATOR
                                        LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
                                        XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
                                        MOV     E,A             ;SAVE XOR RESULT IN E
                                        ANI     $10             ;IS BIT #4 OF RESULT SET
                                        MOV     A,E             ;RESTORE XOR RESULT FROM B
                                        JZ      ECC1            ;CONTINUE IF BIT #4 RESET
                                        XRI     $23             ;ELSE-XOR WITH 23
                                ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                        ;
                                        XRA     A               ;CLEAR A
                                        MOV     C,A             ;CLEAR THE TRANSLATE RESULT
                                        LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
                                        CALL    TRANS           ;TRANSLATE THE BITS
                                        MOV     A,C             ;GET THE TRANSLATED RESULT
                                        STA     ECCCHR          ;STORE RESULT
                                        POP     PSW             ;RESTORE THE ACCUMULATOR
                                        POP     D               ;RESTORE D&E
                                        POP     H               ;RESTORE H&L
                                        RET

                                ECCTBL: $08                     ;BIT 0 = POSITION 3
                                        $20                     ;BIT 1 = POSITION 5
                                        $02                     ;BIT 2 = POSITION 1
                                        $40                     ;BIT 3 = POSITION 6
                                        $80                     ;BIT 4 = POSITION 7
                                        $01                     ;BIT 5 = POSITION 0
                                        $10                     ;BIT 6 = POSITION 4
                                        $04                     ;BIT 7 = POSITION 2

                                ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;
48E3  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
48E5  7B              4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
48E6  A0              4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
48E7  CA  ED  48     10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
48EA  79              4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
48EB  B6              7.0          ORA   M            ;'OR' IN NEW POSITION
48EC  4F              4.0          MOV   C,A          ;SAVE RESULT

48ED  78              4.0  TRANS2: MOV   A,B
48EE  07              4.0          RLC                ;POSITION MASK TO NEXT BIT
48EF  47              4.0          MOV   B,A
48F0  D8             12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
48F1  23              6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
48F2  C3  E5  48     10.0          JMP   TRANS1       ;PROCESS NEXT BIT
```

```
                                .SBTTL  SUBROUTINE TRK
48F5                            SSUB
                                ;*****************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*GENERATE BAD TRACK TRANSLATION
48F5                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE GENRATES THREE TRACK NUMBERS FOR THE BADT4X5 SUBROUTINE
                                ;*BASED ON THE VALUE OF INDAT AND THREE TIE BUS TIMES
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND BITS SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITIONS
                                ;*          DATA                WITH ECC CORRECTION
                                ;*      -------------           -------------------
                                ;*
                                ;*         XXXX0000                    0,1,2
                                ;*         XXXX0001                    1,2,3
                                ;*         XXXX0010                    2,3,4
                                ;*         XXXX0011                    3,4,5
                                ;*         XXXX0100                    4,5,6
                                ;*         XXXX0101                    5,6,7
                                ;*         XXXX0110                    6,7,P
                                ;*         XXXX0111                    7,P,0
                                ;*         XXXX1000                    P,0,1
                                ;*         XXXX1001                    1,2,3
                                ;*         XXXX1010                    2,3,4
                                ;*         XXXX1011                    3,4,5
                                ;*         XXXX1100                    4,5,6
                                ;*         XXXX1101                    5,6,7
                                ;*         XXXX1110                    6,7,P
                                ;*         XXXX1111                    7,P,0
                                ;*
                                ;*      X = DON'T CARE BITS
48F5                            S
                                ;*****************************************************************************

48F5  3A 54 49      13.0  TRK:  LDA   INDAT         ;GET INPUT DATA
48F8  E6 0F          7.0        ANI   $0F           ;REMOVE 4 BITS
48FA  FE 08          7.0        CPI   $8            ;PARITY TRACK?
48FC  CA 01 49      10.0        JZ    TRKA          ;YES
48FF  E6 07          7.0        ANI   $7            ;NO-IGNORE BIT3
4901  3C             4.0  TRKA: INR   A             ;INCREMENT FOR SUBROUTINE
4902  32 B1 48      13.0        STA   BAD1          ;SAVE IT.
4905  3D             4.0        DCR   A             ;PREPARE TO USE AS TABLE INDEX
4906  21 46 49      10.0        LXI   H,TIMTBL      ;INCORRECT TIE BUS TIME TABLE START
4909  5F             4.0        MOV   E,A           ;LOW HALF OF TABLE INDEX
490A  16 00          7.0        MVI   D,0           ;ZERO HIGH HALF
490C  19            10.0        DAD   D             ;PRODUCE TABLE ENTRY ADDRESS
```

```
2757  490D  7E              7.0          MOV     A,M             ;GET INCORRECT TIE BUS TIME FROM TABLE
2758  490E  32  50  49       13.0         STA     TIET1           ;SAVE FIRST TIE BUS JAM TIME
2759  4911  3A  B1  48       13.0         LDA     BAD1
2760  4914  3C               4.0          INR     A               ;GENERATE VALUE FOR BAD2
2761  4915  FE  0A           7.0          CPI     $A              ;TOO BIG?
2762  4917  C2  1C  49       10.0         JNZ     TRKB            ;JUMP IF OK
2763  491A  3E  01           7.0          MVI     A,@1            ;YES
2764  491C  32  B2  48       13.0  TRKB:  STA     BAD2            ;SAVE IT
2765  491F  3D               4.0          DCR     A               ;PREPARETO USE AS TABLF INDEX
2766  4920  21  46  49       10.0         LXI     H,TIMTBL        ;TIE BUS TIME TABLE START
2767  4923  5F               4.0          MOV     E,A             ;LOW HALF OF TABLE INDEX
2768  4924  16  00           7.0          MVI     D,0             ;CLEAR UPPER HALF
2769  4926  19               10.0         DAD     D               ;PRODUCE TABLE ENTRY ADDRESS
2770  4927  7E               7.0          MOV     A,M             ;GET SECOND TIE BUS JAM TIME
2771  4928  32  51  49       13.0         STA     TIET2           ;SAVE IT
2772  492B  3A  B2  48       13.0         LDA     BAD2
2773  492E  3C               4.0          INR     A
2774  492F  FE  0A           7.0          CPI     $A
2775  4931  C2  36  49       10.0         JNZ     TRKC
2776  4934  3E  01           7.0          MVI     A,@1
2777  4936  32  B3  48       13.0  TRKC:  STA     BAD3
2778  4939  3D               4.0          DCR     A
2779  493A  21  46  49       10.0         LXI     H,TIMTBL
2780  493D  5F               4.0          MOV     E,A
2781  493E  16  00           7.0          MVI     D,0
2782  4940  19               10.0         DAD     D
2783  4941  7E               7.0          MOV     A,M
2784  4942  32  52  49       13.0         STA     TIET3
2785  4945  C9               10.0         RET
2786  4946  02                    TIMTBL: .BYTE   2               ; TIE BUS TIME FOR BIT 0
2787  4947  03                            .BYTE   3               ; TIE BUS TIME FOR BIT 1
2788  4948  00                            .BYTE   0               ; TIE BUS TIME FOR BIT 2
2789  4949  01                            .BYTE   1               ; TIE BUS TIME FOR BIT 3
2790  494A  04                            .BYTE   4               ; TIE BUS TIME FOR BIT 4
2791  494B  08                            .BYTE   8               ; TIE BUS TIME FOR BIT 5
2792  494C  07                            .BYTE   7               ; TIE BUS TIME FOR BIT 6
2793  494D  05                            .BYTE   5               ; TIE BUS TIME FOR BIT 7
2794  494E  06                            .BYTE   6               ; TIE BUS TIME FOR BIT P
2795
2796
```

```
2798                                    .SBTTL  PROGRAM VARIABLES
2799
2800  494F   00                         TEMPX:  .BYTE   0        ;TEMP STORAGE
2801  4950   00                         TIET1:  .BYTE   0        ;FIRST TIE JAM TIME
2802  4951   00                         TIET2:  .BYTE   0        ;SECOND TIE BUS JAM TIME
2803  4952   00                         TIET3:  .BYTE   0        ;THIRD TIE BUS JAM TIME
2804  4953   00                         SUNIT:  .BYTE   0        ;TU PORT #
2805  4954   00                         INDAT:  .BYTE   0        ;DATA BYTE TO ECC
2806  4955   00                         INDATP: .BYTE   0        ;DATA BYTE PARITY
2807  4956   00                         UNITMP: .BYTE   0        ;UNIT MAP
2808  4957   00   00                    AMTMSK: .WORD   0        ;AMTIE MASK WORD
2809  4959   00                         DATAA:  .BYTE   0        ;ACTUAL DATA AFTER TRANSLATION
2810  495A   00                         DATAAP: .BYTE   0        ;ACTUAL DATA PARITY AFTER TRANSLATION
2811  495B   00                         DATAE:  .BYTE   0        ;EXPECTED DATA AFTER TRANSLATION
2812  495C   00                         DATAEP: .BYTE   0        ;EXPECTED DATA PARITY AFTER TRANSLATION
2813  495D 0007                         BUFFER: .BLKB   7        ;DATA BYTE STORAGE
2814       0000                                 .END
```

```
A      =%0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
AMTMSK  4957    ARAIDF= 0098    ASAVE   4F9B    ATTCD   4F97
AXNUM   4F91    B      =%0000   BAD     47DD    BADST = 0090
BADT4X  47CE    BAD1    48B1    BAD1C   482A    BAD2    48B2
BAD2C   4835    BAD3    48B3    BAD3C   4847    BIT0  = 0001
BIT1  = 0002    BIT15 = 8000    BIT2  = 0004    BIT3  = 0008
BIT4  = 0010    BIT5  = 0020    BIT6  = 0040    BIT7  = 0080
BIT8  = 0100    BIT9  = 0200    BRKPBC= 4F0A    BRKRAM= 4F10
BRKSTR= 4E60    BRKXCT= 4F00    BSAVE   4F9C    BUFFER  495D
BYTCNT= 00D4    BYTEH   4F24    BYTEL   4F23    B4X5    47EE
C      =%0001   CASCT   4F21    CASCTL= 00A0    CASSTA= 00A0
CATTH = 0089    CATTL = 0088    CBUSST= 00A1    CBYTH = 008B
CBYTL = 008A    CDG1H = 0087    CDG1L = 0086    CDG2H = 0093
CDG2L = 0092    CDG3H = 0095    CDG3L = 0094    CDVTH = 008D
CDVTL = 008C    CHPTIE= 0028    CHOTIE= 0020    CH1TIE= 0021
CH2TIE= 0022    CH3TIE= 0023    CH4TIE= 0024    CH5TIE= 0025
CH6TIE= 0026    CH7TIE= 0027    CKLOP = 2815    CLEAR   47A3
CLECC   48B4    CLKCTL= 00F0    CLOCK   4F26    CLRLP   47A7
CMC0H - 0099    CMC0L = 0098    CMC1H = 009B    CMC1L = 009A
CMC2H = 009D    CMC2L = 009C    CMC3H = 009F    CMC3L = 009E
CMINH = 0097    CMINL = 0096    CNTCTL= 00D7    CRCWRD= 0018
CSAVE   4F9D    CSRLH = 0091    CSRLL = 0090    CTCH  = 0085
CTCL  - 0084    CTSTH = 008F    CTSTL = 008E    CXCTH = 0081
CXCTL = 0080    CXINH = 0083    CXINL = 0082    C.    = 0001
C.AVAI= 0080    C.DP  = 0008    C.DSE = 0010    C.DTU = 0003
C.DVA = 0008    C.FAIL= 00FC    C.FMT = 0070    C.FNCT= 003E
C.GO  = 0001    C.INTC= 00FE    C.MAIN= 0020    C.NSA = 0080
C.RCT = 00FC    C.SER = 0080    C.SHR = 0040    C.SKPC= 000F
C.TAPE- 0040    C.WCS = 0002    C4X5    47F7    D      =%0002
DATAA   4959    DATAAP  495A    DATACT- 00D0    DATAE   495B
DATAEP  495C    DBUS    4F28    DBUSCT= 00C0    DBUSST= 00C0
DDRA  - 00D8    DDRAIN= 0010    DDRB  = 00D9    DDRBIN= 0002
DDRC  = 00DA    DDRCIN= 0001    DDRCO - 0088    DDRCTL= 00DB
DIAFLG  4F22    DIAGPG= 4300    DIAGRM= 4F90    DIARD = 000B
DONE1 - 0045    DONINT= 0010    DSAVE   4F9E    DSE   = 0006
DUMMU   467A  G DUMMV   46B3  G DUMMW   4454  G DUMMX   44AA  G
DUMMY   431E  G D.ATH0- 0001    D.ATH1= 0002    D.EOTD- 0010
D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004    D.TACH= 0008
D.WR4 = 0080    D4X5    47E6    E      =%0003   ECC     48B9
ECCBAD- 0042    ECCCHR  48E2    ECCCOR= 0019    ECCOK = 0041
ECCSTA= 001A    ECCTBL  48DA    ECCTST- 000E    ECC1    48C9
EDATA = 0095    EOTCLR= 0003    ERFLG   4F93    ERLP  = 2809
ERLPA - 280F    ERLPB = 2812    ERLPE - 280C    ERNUM   4F90
ERRCNT= 00D6    ESAVE   4F9F    EXIT    4795    E.ACRC= 00?0
E.AMT = 0020    E.CDP = 0080    E.CRC = 0080    E.PNTR= 0008
E.RPE - 0040    E.STEC= 0001    E.TTEC= 0002    E.UNC = 0004
E4X5    4810    FIFORD= 006A    FORMAT  4F25    FOUND   4344
FWDTST- 0061    F4X5    485C    GCRID = 0089    GCRSET= 0002
GOODTM- 0092    GP4X5   48AF    H      -%0004   HLSAVE  4FA0
IE      0008    INDAT   4954    INDATP  4955    INTSTA= 00E0
INV4X5  4840    ITERA   4F9A    I.PWR = 0020    I.RMPE= 0040
I5.5  = 0010    I6.5  = 0020    I7.5  = 0040    KCALL = 005F
KCLR  = 007B    KDEP  = 003F    KENAB = 0078    KEXAM - 003E
KEYBRD= 00C8    KEY1  = 0078    KEY10 - 006D    KEY11 = 006E
```

```
KEY12 = 006F    KEY13 = 005C    KEY14 = 005D    KEY15 = 005E
KEY16 = 005F    KEY17 = 003C    KEY18 = 003D    KEY19 = 003E
KEY2  = 0079    KEY20 = 003F    KEY3  = 007A    KEY4  = 007B
KEY5  = 0074    KEY6  = 0075    KEY7  = 0076    KEY8  = 0077
KEY9  = 006C    KINTA = 006F    KLDAD = 003D    KN0   = 003C
KN1   = 005C    KN2   = 005D    KN3   = 005E    KN4   = 006C
KN5   = 006D    KN6   = 006E    KN7   = 0074    KN8   = 0075
KN9   = 0076    KU2   = 0079    KU3   = 007A    KU8   = 0077
L     =%0005    LBLANK= 000F    LCE   = 0008    LCH   = 000C
LCL   = 000D    LCP   = 000E    LC0   = 0000    LC1   = 0001
LC2   = 0002    LC3   = 0003    LC4   = 0004    LC5   = 0005
LC6   = 0006    LC7   = 0007    LC8   = 0008    LC9   = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94
LPNUM   4F92    M     =%0006    MBSEL = 00E0    MB.A  = 0008
MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A    MM    = 8000
MSE   = 0008    MSGN  = 000F    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR- 0004    MT.MOT= 0002    MT.NWI= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT- 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT- 0080    M.DEM = 0020    M.EBL = 0010    M.EXC = 0008
M.FAIL= 0008    M.ILR - 0010    M.INIT= 0010    M.OCC = 0020
M.ONLI- 0001    M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE- 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040
M.UNIT  0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5    0001    M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088
OKAY    00FF    OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040
PADCNT- 00D5    PADCRC= 0080    PDIAG = 0048    PEID  = 008A
PENAB   004C    PESET = 0001    PL    = 00B1    POINTB  488D
POINTE  4879    PO4X5   47FD    PRDD  = 004C    PRENF = 009C
PS    - 00B2    PSTAT = 0048    PSW   =%0009    P.AMTP= 0001
P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040
P.LWR - 0020    P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002
P.RPOE- 0020    P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010
P.SING- 0080    P.STAT= 0002    P.STPE= 0080    P.TACH= 0008
P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001
P.WPEN- 0010    P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008
P.WP3E- 0004    P.5VOK= 0002    QUE   = 281B    QUEM  = 281E
RAMT  - 0010    RARA  = 0006    RARAI = 0004    RCHBD0= 0048
RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C    RCLRT = 000D
RCMD  = 000B    RCMLP - 0003    RCONT = 0080    RDATA = 0017
RDCLK = 0010    RDON  - 0011    READG = 0007    REND  = 0014
REQST = 2806    RESCHR= 00D1    RETRY   4798    REVTST= 0064
REWIND- 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 000C    RMCTST= 0008
RMK2  - 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK - 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  - 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL- 0005    R.AMT = 0001    R.BOP - 0008
```

```
R.DATA= 0040        R.DON = 0002        R.DRDY= 0010        R.END = 0010
R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008        R.PLOD- 0008
R.PLOO= 0010        R.PL01= 0020        R.POST= 0020        R.STNM= 0002
R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040
R.VOK = 0080        R00H  = 0081        R00L  = 0080        R01H  = 0083
R01L  = 0082        R02H  = 0085        R02L  = 0084        R03H  = 0087
R03L  = 0086        R04H  = 0089        R04L  = 0088        R05H  = 008B
R05L  = 008A        R06H  = 008D        R06L  = 008C        R07H  = 008F
R07L  = 008E        R10H  = 0091        R10L  = 0090        R11H  = 0093
R11L  = 0092        R12H  = 0095        R12L  = 0094        R13H  = 0097
R13L  = 0096        R14H  = 0099        R14L  = 0098        R15H  = 009B
R15L  = 009A        R16H  = 009D        R16L  = 009C        R17H  = 009F
R17L  = 009E        R7.5  = 0010        SELCLR= 0000        SETATA= 00A1
SID   = 0080        SOD   = 0080        SOE   = 0040        SP    =%0008
SSCLK = 0040        SSTEP = 0005        STACK = 4FFF        STATRM= 4F20
STPCT   4F20        STRSP = 5000        SUNIT   4953        TAB4X5  4891
TADR00= 0080        TADR01= 0081        TADR02= 0082        TADR03= 0083
TADR04= 0084        TADR05= 0085        TADR06= 0086        TADR07= 0087
TADR10= 0088        TADR11= 0089        TADR12= 008A        TADR13= 008B
TAMT  = C044        TASEL = 0080        TCMD  = 0040        TC.FWD= 0040
TC.INH= 0008        TC.LWR= 0004        TC.REV= 0020        TC.WRT= 0010
TEMP    4F99        TEMPX   494F        TEST1   4300        TEST2   4573
TIET1   4950        TIET2   4951        TIET3   4952        TIMTBL  4946
TMF   = 0099        TMRDY = 0040        TRANS   48E3        TRANS1  48E5
TRANS2  48ED        TRK     48F5        TRKA    4901        TRKB    491C
TRKC    4936        TRKENA= 00D2        TRNIN   48A1        TRNOUT  48A5
TRNTMP  48B0        TSET  = 2803        TSTEND= 2818        TSTS  = 0040
TST01A  445D        TST01B  4461        TST01C  437F        TST01F  439A
TST01G  43EB        TST01H  4423        TST01I  43AD        TST01K  44B3
TST01L  44C7        TST01M  44DA  G     TST01N  451C        TST01Q  4501
TST01R  44F8  G     TST01S  4511        TST01T  450B  G     TST01U  4533  G
TST01V  454F  G     TST01W  43B9        TST01X  4351        TST01Y  44B8
TST01Z  43C0        TST02A  4683        TST02B  4687        TST02C  45B0
TST02F  45CB        TST02G  4618        TST02H  4649        TST02I  45DE
TST02K  46BC        TST02L  46D3        TST02M  46EA  G     TST02N  4734
TST02Q  4715        TST02R  470C  G     TST02S  4729        TST02T  4723  G
TST02U  4751  G     TST02V  4773  G     TST02W  45EA        TST02X  457D
TST02Y  46C1        TST02Z  45F1        TUSEL0= 00D1        TUSEL1= 00D2
TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002        T.BOT = 0004
T.EOT - 0002        T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020
T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008        T.PS1J= 0010
T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010        T.SCLK= 0002
T4X5    47D2        UIBG  = 00A1        UNITMP  4956        UNITSL  4327
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL- 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X     =%000A
X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002        X.ROME= 0001
X.WCLK= 0001        Y     -%000B        .     - 4964
```

ERRORS DETECTED: 0

*ECC5.A78/PTP,ECC5=NLIST,PARAM,MACRO,LIST,ECC5
RUN-TIME: 5 7 0 SECONDS
CORE USED: 10K

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:11 SEQ 1059
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  CCM1 - CHECK CHARACTER MICRO CONTROLLER PART #1
                                ;ID     CCM1-CHECK CHARACTER MICRO CONTROLLER
                                .SBTTL  TEST 1 - CHECK CHARACTER MICRO INITIAL STATUS TEST
4300                            ST
                                ; **********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*CCM - INITIAL STATUS CHECK
4300                            SD
                                ; **********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CLEARS THE CHECK CHARACTER CONTROLLER AND VERIFIES THAT THE
                                ;*ACRC NOT OK AND CRC NOT OK ARE BOTH SET.
4300                            SP
                                ; **********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   SET NORMAL READ PATH CLOCK
                                ;*   PUT THE MACHINE IN GCR WRITE MODE
                                ;*   CALL SUBROUTINE CLEAR
                                ;*   ENABLE THE READ PATH CLOCK
                                ;*   SET PLO BYPASS MODE 2
                                ;*   ISSUE CLEAR ALL COMMAND
                                ;*   WAIT
                                ;*   STOP THE READ PATH
                                ;*   INPUT THE ACRC AND CRC NOT OK STATUS BITS
                                ;*   IF ACRC NOT OK=1
                                ;*   :   THEN-CONTINUE
                                ;*   :   ELSE-ERROR
                                ;*   ENDIF
                                ;*   IF CRC NOT OK=1
                                ;*   :   THEN-CONTINUE
                                ;*   :   ELSE-ERROR
                                ;*   ENDIF
                                ;*ENDTST
4300                            SE
                                ; **********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*CCM1 MICRO TEST 01
                                ;*CCM1 MICRO ERROR 01
                                ;*CCM1-CHECK CHARACTER CONTROLLER-INITIAL STATUS TEST
                                ;*M8952, M8953
                                ;*ACRC NOT OK = 0 AFTER CLEAR SHOULD = 1
                                ;*
                                ;*CCM1 MICRO TEST 01
                                ;*CCM1 MICRO ERROR 02
                                ;*CCM1-CHECK CHARACTER CONTROLLER-INITIAL STATUS TEST
                                ;*M8952, M8953
                                ;*CRC NOT OK = 0 AFTER CLEAR SHOULD = 1
4300                            S
                                ; **********************************************************************
```

```
                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
(1)  4300  3E  01              7.0                      MVI     A,@1            ;DEFINE THE TEST NUMBER
(1)  4302  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
                                        ;%CCM1-CHECK CHARACTER CONTROLLER-INITIAL STATUS TEST
                                        ;&M8952, M8953, M8951

4305  3E  10                   7.0  TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
4307  D3  F0                  10.0           OUT     CLKCTL
4309  3E  18                   7.0           MVI     A,W.GCR.W.WRITE ;PUT THE MACHINE IN GCR
430B  D3  D3                  10.0           OUT     WMCCTL          ;WRITE MODE
430D  CD  8E  45              18.0           CALL    CLEAR           ;CALL SUBROUTINE CLEAR
4310  3E  20                   7.0           MVI     A,R.PLO1        ;SET PLO BYPASS MODE 2 AND
4312  D3  09                  10.0           OUT     RPCTL           ;ENABLE THE READ PATH CLOCK
4314  3E  0D                   7.0           MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
4316  D3  08                  10.0           OUT     RCMD
4318  00                       4.0           NOP                     ;
4319  00                       4.0           NOP
431A  00                       4.0           NOP
431B  00                       4.0           NOP
431C  00                       4.0           NOP
431D  3E  21                   7.0           MVI     A,R.PLO1.R.STPC ;STOP THE READ PATH
431F  DB  1A                  10.0           IN      ECCSTA          ;GET THE CCM STATUS
4321  E6  10                   7.0           ANI     E.ACRC          ;ACRC NOT OK=1?
4323  C2  2B  43              10.0           JNZ     T01CN1          ;YES-CONTINUE
4326                                         ERR     TST01X,T01CN1
(1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1)  4326  CD  09  28         18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)        0001                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1)  4329  01                                .BYTE   MSGN            ;MESSAGE NUMBER ID
(1)  432A  00                                .BYTE
(1)  432B  CD  15  28         18.0  T01CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1)  432E  DA  05  43         10.0                      JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>ACRC NOT OK = 0 AFTER CLEAR SHOULD  1
4331  DB  1A                  10.0           IN      ECCSTA
4333  E6  80                   7.0           ANI     E.CRC
4335  C2  3D  43              10.0           JNZ     T01CN2
4338                                         ERR     TST01X,T01CN2
(1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1)  4338  CD  09  28         18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)        0002                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1)  433B  02                                .BYTE   MSGN            ;MESSAGE NUMBER ID
(1)  433C  00                                .BYTE
(1)  433D  CD  15  28         18.0  T01CN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1)  4340  DA  05  43         10.0                      JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>CRC NOT OK = 0 AFTER CLEAR SHOULD - 1
```

```
1401  4343                                      ENDTST  TST01X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4343                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4343  CD  06  28        18.0                            CALL    REQST
 (2)  4346  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4347  00  00                                            .WORD                   ;SYSTEM '#' COUNT
 (2)  4349  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  434B  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  434C  07                                                .BYTE   7               ;REQUEST CODE
 (1)  434D  3A  9A  4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  4350  3D                 4.0            DCR     A               ;DOWNCOUNT
 (1)  4351  32  9A  4F        13.0            STA     ITERA           ;SAVE COUNT
 (1)  4354  F2  05  43        10.0            JP      TST01X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - CRC AND ACRC TEST
1404  4357                      ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*CCM CRC AND ACRC TEST
1406  4357                      SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WRITES RECORDS FROM 1 - 128 BYTES AND CHECKS THAT THE CHECK
                                ;*CHARACTER MICRO-CONTROLLER WILL ACCURATFLY RECOMPUTE THE ACRC AND CRC
                                ;*CHARACTERS.
1410  4357                      SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*   GET USER SELECTED TU PORT
                                ;*   CLEAR THE BYTE COUNT CONTROL BYTE (SOFTWARE)
                                ;*   BGNDO
                                ;*   :   CALL SUBROUTINE CLEAR
                                ;*   :   CLEAR THE TU PORT COMMAND REGISTER
                                ;*   :   ENABLE THE WRITE AND READ PATHS TO THE SELECTED TU PORT
                                ;*   :   CLEAR THE XMC TRACK ENABLE LINES
                                ;*   :   CLEAR THE TU PORT AMTIE LINES
                                ;*   :   SET NORMAL READ PATH CLOCKS
                                ;*   :   PUT THE MACHINE IN GCR MODE
                                ;*   :   SET THE TU PORT IN LOOP WRITE/READ MODE
                                ;*   :   SET THE READ PATH CONTROL REGISTER TO BYPASS 2, TIE JAM, AND PLO DISABLE
                                ;*   :   CLEAR THE READ CHANNEL TIE BUS
                                ;*   :   CLOCK THE READ CHANNEL FIFO'S
                                ;*   :   ISSUE READ PATH CLEAR ALL COMMAND
                                ;*   :   WAIT
                                ;*   :   SET THE READ PATH CONTROL REGISTER TO BYPASS 2, TIE JAM, PLO DISABLE AND STOP
                                ;*   :   INITIALIZE THE DDR CONTROL
                                ;*   :   SET NORMAL WRITE PATH CLOCKS
                                ;*   :   SET THE WMCCTL REGISTER TO RESTART
                                ;*   :   WAIT
                                ;*   :   CLEAR THE WMCCTL REGISTER
                                ;*   :   WAIT 1280 MICROSECONDS
                                ;*   :   SETUP SINGLE STEP WRITE CLOCK NORMAL READ CLOCK
                                ;*   :   SELECT THE BYTE, PAD AND ECODE COUNTERS
                                ;*   :   CLEAR THE BYTE, PAD AND ECODE COUNTERS
                                ;*   :   INCREMENT THE BYTE COUNT CONTROL BYTE
                                ;*   :   INITIALIZE THE DDR FOR A DATA TRANSFER
                                ;*   :   CALL OPERATIONAL MICROCODE MOD7 ROUTINE
                                ;*   :   CALCULATE THE MOD32 BITS
                                ;*   :   CALCULATE THE RESIDUAL CHARACTER COUNT
                                ;*   :   CALCULATE THE PAD CHARACTER COUNT
                                ;*   :   CALCULATE THE PAD CRC BIT
                                ;*   :   LOAD THE IMAGE MODE OF DATA TRANSFER
                                ;*   :   ENABLE ALL XMC TRACK ENABLE BITS
                                ;*   :   SET ALL THE AMTIE LINES
```

```
                                ;*  :  SET THE READ PATH CONTROL REGISTER TO BYPASS 1, PLO DISABLE AND STOP
                                ;*  :  CLEAR THE TIE BUS
                                ;*  :  CLEAR THE SCLK LATCH (SOFTWARE)
                                ;*  :  CLEAR THE XL WRITE CLOCK LATCH (SOFTWARE)
                                ;*  :  SET THE WRITE PATH CONTROL REGISTER TO WMC ENAB, XMC ENAB, WRITE, GCR
                                ;*  :  SET THE READ PATH CONTROL REGISTER TO READ AFTER WRITE GCR
                                ;*  :  BGNDO
                                ;*  :  :  SET THE LOOP COUNT TO 2
                                ;*  :  :  BGNDO
                                ;*  :  :  :  CALL SUBROUTINE CLKSYS
                                ;*  :  :  :  GET THE DBUS STATUS REGISTER
                                ;*  :  :  :  IF SCLOCK BIT=1
                                ;*  :  :  :  :  THEN-IF SCLK LATCH=1
                                ;*  :  :  :  :  :       THEN-CONTINUE
                                ;*  :  :  :  :  :       ELSE-CALL SUBROUTINE SETDDR
                                ;*  :  :  :  :  :       :       SET THE SCLK LATCH
                                ;*  :  :  :  :  :       ENDIF
                                ;*  :  :  :  :  :  ELSE-CLEAR THE SCLK LATCH
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  GET THE WMC STATUS
                                ;*  :  :  :  :  :  IF TRANSLATOR WRITE CLOCK BIT=1
                                ;*  :  :  :  :  :  :  THEN-IF XL WRITE CLOCK LATCH=1
                                ;*  :  :  :  :  :  :        :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :        :  ELSE-CLEAR THE AMTIE LINES
                                ;*  :  :  :  :  :  :        :       SET THE XL WRITE CLOCK LATCH
                                ;*  :  :  :  :  :  :        ENDIF
                                ;*  :  :  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  DECREMENT THE LOOP COUNT
                                ;*  :  :  :  :  :  :  DO UNTIL THE LOOP COUNT=0
                                ;*  :  :  :  :  :  ENDDO
                                ;*  :  :  :  :  :  SINGLE STEP THE READ PATH
                                ;*  :  :  :  :  :  GET THE WRITE MICRO STATUS
                                ;*  :  :  :  :  :  DO UNTIL THE WMC NOT DONE AND XMC NOT DONE STATUS BITS=0
                                ;*  :  :  :  :  ENDDO
                                ;*  :  :  :  :  SET THE LOOP COUNT TO -2000
                                ;*  :  :  :  :  BGNDO
                                ;*  :  :  :  :  :  CALL SUBROUTINE CLKSYS
                                ;*  :  :  :  :  :  CALL SUBROUTINE CLKSYS
                                ;*  :  :  :  :  :  SINGLE STEP THE READ PATH
                                ;*  :  :  :  :  :  INCREMENT THE LOOP COUNT
                                ;*  :  :  :  :  :  DO UNTIL READ CHANNEL STATUS VALID=1 OR LOOP COUNT=0
                                ;*  :  :  :  :  ENDDO
                                ;*  :  :  :  :  IF LOOP COUNT=0
                                ;*  :  :  :  :  :  THEN-ERROR
                                ;*  :  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  :  ENDIF
                                ;*  :  :  :  :  IF ECC STATUS BIT CRC ERROR 1
                                ;*  :  :  :  :  :  THEN-ERROR
                                ;*  :  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  :  ENDIF
                                ;*  :  :  :  :  IF ECC STATUS BIT ACRC ERROR-1
                                ;*  :  :  :  :  :  THEN-ERROR
                                ;*  :  :  :  :  :  ELSE-CONTINUE
```

```
1502                                    ;*   :  ENDIF
1503                                    ;*   :  INCREMENT THE BYTE COUNT CONTROL BYTE
1504                                    ;*   :  DO UNTIL THE BYTE COUNT CONTROL BYTE=1
1505                                    ;*  ENDDO
1506                                    ;*ENDTST
1507  4357                              SE
 (1)                                    ; ****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1508                                    ;*CCM1 MICRO TEST 02
1509                                    ;*CCM1 MICRO ERROR 03
1510                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1511                                    ;*M8952
1512                                    ;*OPERATOR ERROR NO TM78 UNITS SELECTED
1513                                    ;*FATAL ERROR - TEST ABORTED
1514                                    ;*
1515                                    ;*CCM1 MICRO TEST 02
1516                                    ;*CCM1 MICRO ERROR 04
1517                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1518                                    ;*M8952
1519                                    ;*READ CHANNEL ''STATUS VALID'' NOT SET AFTER XMC DONE
1520                                    ;*FATAL ERROR - TEST ABORTED
1521                                    ;*BYTE-SCLK COUNT = LLL
1522                                    ;*DATA FORMAT - MM
1523                                    ;*SKIP COUNT = NN
1524                                    ;*
1525                                    ;*CCM1 MICRO TEST 02
1526                                    ;*CCM1 MICRO ERROR 05
1527                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1528                                    ;*M8952
1529                                    ;*CRC ERROR AFTER DATA TRANSFER
1530                                    ;*BYTE-SCLK COUNT = LLL
1531                                    ;*DATA FORMAT = MM
1532                                    ;*SKIP COUNT - NN
1533                                    ;*
1534                                    ;*CCM1 MICRO TEST 02
1535                                    ;*CCM1 MICRO ERROR 06
1536                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1537                                    ;*M8952
1538                                    ;*ACRC ERROR AFTER DATA TRANSFER
1539                                    ;*BYTE-SCLK COUNT = LLL
1540                                    ;*DATA FORMAT = MM
1541                                    ;*SKIP COUNT = NN
1542  4357                              S
 (1)                                    ; ****************************************************************************
1543  4357                              TST2:   TESTX   @2
 (1)  4357  3E   02          7.0                         MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4359  CD   03   28    18.0                         CALL    TSET            ;SETUP THE TEST
1544                                    ;%CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1545                                    ;&M8952
1546                                    ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
1547                                    ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.
1548
1549  435C                                      REQ     @7,0,0,0
```

```
(1) 435C CD 06 28 18.0                          CALL    REQST
(1) 435F 00                                     .BYTE   0               ;DATA PATTERN NUMBER
(1) 4360 00 00                                  .WORD   0               ;SYSTEM "0" COUNT
(1) 4362 00 00                                  .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(1) 4364 00                                     .BYTE                   ;DATA COMPARE FLAG IF -1
(1) 4365 07                                     .BYTE   @7              ;REQUEST CODE
1550 4366                               RIN     R12L
(1) 4366 DB 94              10.0                IN      R12L            ;READ R12L INTO AC
(1) 4368 7F                  4.0                MOV     A,A             ;RETRY LINK
1551 4369 32 CF 45          13.0        STA     UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
1552 436C AF                 4.0 T2LPB:  XRA     A
1553 436D 32 CD 45          13.0        STA     TPBCNT          ;CLEAR THE TEST PASS/BYTE COUNT
1554 4370 CD 8E 45          18.0 T2LPA: CALL    CLEAR
1555 4373 06 00              7.0        MVI     B,0
1556 4375 3A CF 45          13.0        LDA     UNITMP          ;GET SELECTED UNIT MAP
1557 4378 E6 01              7.0        ANI     $01             ;UNIT 0 PRESENT?
1558 437A C2 A6 43          10.0        JNZ     FOUND           ;YES-USE IT
1559 437D 04                 4.0        INR     B
1560 437E 3A CF 45          13.0        LDA     UNITMP          ;
1561 4381 E6 02              7.0        ANI     $02             ;UNIT 1 PRESENT?
1562 4383 C2 A6 43          10.0        JNZ     FOUND           ;YES-USE IT
1563 4386 04                 4.0        INR     B
1564 4387 3A CF 45          13.0        LDA     UNITMP          ;
1565 438A E6 04              7.0        ANI     $04             ;UNIT 2 PRESENT?
1566 438C C2 A6 43          10.0        JNZ     FOUND           ;YES-USE IT
1567 438F 04                 4.0        INR     B
1568 4390 3A CF 45          13.0        LDA     UNITMP
1569 4393 E6 08              7.0        ANI     $08             ;UNIT 3 PRESENT
1570 4395 C2 A6 43          10.0        JNZ     FOUND           ;YES-USE IT
1571 4398                               ERR     EXIT,DUMMY
(1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 4398 CD 09 28           18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)      0003                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 439B 03                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 439C 00                                 .BYTE
(1) 439D CD 15 28           18.0        DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1) 43A0 DA 4F 45           10.0                JC      EXI             ;LOOP ADDRESS IF LOOP SPECIFIED
1572                                    ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1573                                    ;>FATAL ERROR - TEST ABORTED
1574 43A3 C3 4F 45          10.0        JMP     EXIT
1575                                    ;SET UP FOR THE DATA TRANSFER
1576 43A6 DB E0             10.0 FOUND: IN      INTSTA
1577 43A8 E6 80              7.0        ANI     BIT7
1578 43AA B0                 4.0        ORA     B
1579 43AB D3 E0             10.0        OUT     MBSEL
1580 43AD AF                 4.0        XRA     A
1581 43AE D3 40             10.0        OUT     TCMD            ;CLEAR TU COMMAND REGISTER
1582 43B0 3E 14              7.0        MVI     A,P.WPEN!P.RPEN ;ENABLE THE WRITE AND READ PATHS
1583 43B2 D3 4C             10.0        OUT     PENAB
1584 43B4 AF                 4.0        XRA     A
1585 43B5 D3 D2             10.0        OUT     TRKENA          ;CLEAR ALL TRACK ENABLE BITS
1586 43B7 D3 D2             10.0        OUT     TRKENA
1587 43B9 D3 44             10.0        OUT     TAMT            ;CLEAR THE AMTIE LINES
```

```
1589                                    ;RESTART READ MICROCONTROLLER ROM PROGRAM
1590  43BB  3E  10          7.0  TST2L:  MVI   A,RDCLK          ;SET NORMAL READ PATH CLOCKS
1591  43BD  D3  F0         10.0         OUT   CLKCTL
1592
1593  43BF  3E  10          7.0         MVI   A,W.GCR          ;SET MACHINE IN GCR MODE
1594  43C1  D3  D3         10.0         OUT   WMCCTL
1595
1596  43C3  3E  20          7.0         MVI   A,P.LWR          ;SET THE PORT CONTROL TO
1597  43C5  D3  48         10.0         OUT   PDIAG            ;LOOP WRITE/READ
1598
1599  43C7  3E  A8          7.0  TST02C: MVI   A,R.PLO1!R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1600                                                         ;SET TIE BUS JAM AND PLO DISABLE
1601  43C9  D3  09         10.0         OUT   RPCTL
1602  43CB  AF              4.0         XRA   A
1603  43CC  D3  0A         10.0         OUT   RTIEB            ;CLEAR THE TIE BUS
1604  43CE  D3  08         10.0         OUT   RFIFOL           ;CLOCK THE FIFO'S
1605  43D0  3E  0D          7.0         MVI   A,RCLRT          ;ISSUE CLEAR ALL COMMAND
1606  43D2  D3  0B         10.0         OUT   RCMD
1607
1608  43D4  00              4.0         NOP                    ;WAIT
1609  43D5  00              4.0         NOP                    ;WAIT
1610  43D6  00              4.0         NOP
1611  43D7  C0              4.0         NOP
1612  43D8  00              4.0         NOP
1613  43D9  3E  A9          7.0         MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN
1614  43DB  D3  09         10.0         OUT   RPCTL
```

```
1616                                    ;RESTART WRITE MICROCONTROLLER ROM PROGRAM
1617
1618  43DD  3E  89          7.0         MVI   A,@211          ;
1619  43DF  D3  DB         10.0         OUT   DDRCTL
1620  43E1  AF              4.0         XRA   A              ;SET NORMAL WRITE PATH CLOCKS
1621  43E2  D3  F0         10.0         OUT   CLKCTL
1622  43E4  3E  01          7.0         MVI   A,W.RST        ;GET 'RST' BIT
1623  43E6  D3  D3         10.0         OUT   WMCCTL         ;RESTART WMC
1624  43E8  3E  15          7.0         MVI   A,SSTEP+RDCLK  ;SET UP SINGLE STEP WRITE CLOCK
1625  43EA  D3  F0         10.0         OUT   CLKCTL         ;AND +10% READ CLOCK
1626  43EC  32  C5  45     13.0         STA   CCTLWD         ;SAVE COPY
1627  43EF  AF              4.0         XRA   A              ;FINISH THE WMC RESTART
1628  43F0  D3  D3         10.0         OUT   WMCCTL
1629  43F2  3E  30          7.0         MVI   A,@60
1630  43F4  D3  D7         10.0         OUT   CNTCTL         ;SELECT BYTE COUNTER
1631  43F6  3E  70          7.0         MVI   A,@160
1632  43F8  D3  D7         10.0         OUT   CNTCTL         ;SELECT THE PAD COUNTER
1633  43FA  3E  B0          7.0         MVI   A,@260
1634  43FC  D3  D7         10.0         OUT   CNTCTL         ;SELECT ECODE COUNTER
1635
1636                                    ;CLEAR ERROR CODE COUNTER
1637  43FE                              CLRECT               ;CLEAR ECODE COUNTER
 (1)  43FE  AF              4.0                 XRA   A              ;CLEAR THE ACCUMULATOR
 (1)  43FF  D3  D6         10.0                 OUT   ERRCNT         ;CLEAR BITS 7-0
 (1)  4401  D3  D6         10.0                 OUT   ERRCNT         ;CLEAR BITS 15-8
1638  4403                              CLRBCT
 (1)  4403  AF              4.0                 XRA   A              ;CLEAR THE ACCUMULATOR
 (1)  4404  D3  D4         10.0                 OUT   BYTCNT         ;CLEAR BITS 7-0
 (1)  4406  D3  D4         10.0                 OUT   BYTCNT         ;CLEAR BITS 15-8
1639  4408                              CLRPCT
 (1)  4408  AF              4.0                 XRA   A              ;CLEAR THE ACCUMULATOR
 (1)  4409  D3  D5         10.0                 OUT   PADCNT         ;CLEAR BITS 7-0
 (1)  440B  D3  D5         10.0                 OUT   PADCNT         ;CLEAR BITS 15-8
1640
1641                                    ;SET DDR CONTROL TO "OUT" FOR DATA INJECTION INTO BAL LOGIC
1642  440D  3A  CD  45      13.0         LDA   TPBCNT         ;GET THE BYTE COUNT
1643  4410  3C              4.0         INR   A
1644  4411  32  CD  45     13.0         STA   TPBCNT         ;SAVE THE NEW BYTE COUNT TO USE
1645  4414                              ROUT  R05L
 (1)  4414  D3  8A         10.0                 OUT   R05L           ;WRITE AC INTO R05L
 (1)  4416  7F              4.0                 MOV   A,A            ;RETRY LINK
1646  4417  3E  88          7.0         MVI   A,@210         ;SET "OUT" BITS
1647  4419  D3  DB         10.0         OUT   DDRCTL         ;LOAD DDR CONTROL
1648
1649  441B  3A  CD  45      13.0         LDA   TPBCNT         ;GET THE BYTE COUNT TO USE
1650  441E  32  23  4F      13.0         STA   BYTEL          ;SAVE THE BYTE COUNT LOW VALUE
1651  4421  AF              4.0         XRA   A
1652  4422                              ROUT  R05H
 (1)  4422  D3  8B         10.0                 OUT   R05H           ;WRITE AC INTO R05H
 (1)  4424  7F              4.0                 MOV   A,A            ;RETRY LINK
1653  4425                              ROUT  R02L
 (1)  4425  D3  84         10.0                 OUT   R02L           ;WRITE AC INTO R02L
 (1)  4427  7F              4.0                 MOV   A,A            ;RETRY LINK
1654  4428  32  24  4F      13.0         STA   BYTEH          ;CLEAR THE BYTE COUNT HIGH VALUE
```

```
1655  442B  32  D2  45   13.0        STA     DPNUM           ;INITIALIZE THE DATA PATTERN
1656  442E  CD  46  00   18.0        CALL    LKMOD7          ;CALL MOD7 ROUTINE FROM OPR U-CODE
1657  4431  3A  CD  45   13.0        LDA     TPBCNT          ;GET THE BC AGAIN
1658  4434  3D               4.0     DCR     A               ;CALC MOD32
1659  4435  E6  1F           7.0     ANI     @37             ;SAVE LOW 5 BITS
1660  4437  57               4.0     MOV     D,A             ;TEMP SAVE
1661  4438  78               4.0     MOV     A,B             ;GET MOD7 REMAINDER FROM LKMOD7
1662  4439  0F               4.0     RRC                     ;POSITION THE MOD7 NUMBER
1663  443A  0F               4.0     RRC
1664  443B  0F               4.0     RRC
1665  443C  B2               4.0     ORA     D               ;MAKE THE RESID CHAR WORD
1666  443D  D3  D1           10.0    OUT     RESCHR          ;LOAD THE RESID CHAR WORD IN WMC
1667  443F  3A  CD  45   13.0        LDA     TPBCNT          ;GET THE BC AGAIN
1668  4442  90               4.0     SUB     B               ;CALC THE VALUE LOADED INTO THE BYTE COUNTER
1669  4443  3E  06           7.0     MVI     A,6
1670  4445  90               4.0     SUB     B               ;CALC THE PAD COUNTER VALUE
1671  4446  32  CC  45   13.0        STA     IMPADCT         ;SAVE THE PAD COUNT TO USE
1672  4449  3C               4.0     INR     A               ;ADD 1 MORE FOR THE INTEL 8253 CHIP
1673  444A  D3  D5           10.0    OUT     PADCNT          ;LOAD PAD COUNT 7-0
1674  444C  AF               4.0     XRA     A
1675  444D  D3  D5           10.0    OUT     PADCNT          ;LOAD PAD CCOUNTER 15-8
1676  444F  3A  CC  45   13.0        LDA     IMPADCT         ;GET PAD COUNT AGAIN
1677  4452  21  CD  45   10.0        LXI     H,TPBCNT        ;POINT TO BC TO USE
1678  4455  86               7.0     ADD     M               ;CALC # TO SEE IF PADCRC BIT NEEDED
1679  4456  3C               4.0     INR     A               ;IN THE DATACTL WORD
1680  4457  0F               4.0     RRC                     ;LOAD THE PADCRC BIT - IF -1 THEN PAD
1681  4458  E6  80           7.0     ANI     BIT7            ;SAVE ONLY THIS BIT
1682  445A  32  CE  45   13.0        STA     PDCRC           ;SAVE AS A FLAG
1683
1684                                ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1685  445D  3E  50           7.0             MVI     A,$50           ;LOAD THE IMAGE MODE FORMAT CODE
1686  445F                                   ROUT    R02H            ;ALSO SAVE IN CAS REG 02
 (1)  445F  D3  85          10.0                    OUT     R02H            ;WRITE AC INTO R02H
 (1)  4461  7F              4.0                     MOV     A,A             ;RETRY LINK
1687  4462  47               4.0             MOV     B,A             ;SAVE THE FMT DATA
1688  4463  3A  CE  45   13.0             LDA     PDCRC           ;GET THE PAD CRC BIT
1689  4466  B0               4.0             ORA     B               ;ADD IN THE FORMAT NUMBER
1690  4467  D3  D0           10.0            OUT     DATACTL         ;LOAD DATA CONTROL WORD
1691
1692                                ;ENABLE WRITE MICROCONTROLLER - START OF DATA XFR - WRITE
1693  4469  3E  FF           7.0             MVI     A,@377
1694  446B  D3  D2           10.0            OUT     TRKENA          ;ENABLE THE TRACKS FROM
1695  446D  D3  D2           10.0            OUT     TRKENA          ;THE TRANSLATOR
1696  446F  D3  44           10.0            OUT     TAMT            ;SET THE AMTIE LINES
1697  4471  3E  21           7.0             MVI     A,P.LWR.P.AMTP  ;SET THE PARITY AMTIE
1698  4473  D3  48           10.0            OUT     PDIAG           ;TO ONE
1699  4475  3E  19           7.0             MVI     A,R.PLOO!R.PLOD.R.STPC
1700  4477  D3  09           10.0            OUT     RPCTL
```

```
1702  4479  AF                    4.0          XRA     A
1703  447A  D3  0A                10.0         OUT     RTIEB           ;CLEAR THE TIE BUS
1704  447C  32  D1  45             13.0         STA     CLKON           ;CLEAR THE SCLK LATCH
1705  447F  32  D0  45             13.0         STA     XLCLK           ;CLEAR THE XL WRITE CLOCK LATCH
1706  4482  3E  D8                 7.0          MVI     A,W.ENAB!W.WRITE!W.GCR!X.ENAB  ;SELECT 'WMC EN' + 'WRITE'
1707  4484  D3  D3                10.0         OUT     WMCCTL          ;LOAD WRITE MICROCONTROLLER CONTROL
1708  4486  3E  07                 7.0          MVI     A,READG         ;LOAD THE GCR READ COMMAND
1709  4488  D3  0B                10.0         OUT     RCMD            ;START THE READ PATH
1710  448A  0E  02                 7.0  T2LOP1: MVI     C,@2            ;LOAD THE LOOP COUNT
1711  448C  CD  B9  45             18.0 T2LOP4: CALL    CLKSYS          ;SINGLE STEP THE MACHINE
1712  448F  DB  C0                 10.0         IN      DBUSSTA         ;GET DATA BUS STATUS
1713  4491  E6  02                 7.0          ANI     T.SCLK          ;SCLOCK SET?
1714  4493  C2  9D  44             10.0         JNZ     1$              ;YES-GO PROCESS
1715  4496  AF                    4.0          XRA     A               ;NO-CLEAR THE SCLOCK LATCH
1716  4497  32  D1  45             13.0         STA     CLKON
1717  449A  C3  AC  44             10.0         JMP     2$              ;CONTINUE
1718  449D  3A  D1  45             13.0  1$:    LDA     CLKON           ;IF THE SCLOCK LATCH NOT=0
1719  44A0  A7                    4.0          ANA     A               ;
1720  44A1  C2  AC  44             10.0         JNZ     2$              ;THEN-CONTINUE
1721  44A4  CD  52  45             18.0         CALL    SETDDR          ;ELSE-LOAD DATA INTO THE DDR REGISTERS
1722  44A7  3E  FF                 7.0          MVI     A,@377          ;SET THE SCLOCK LATCH
1723  44A9  32  D1  45             13.0         STA     CLKON
1724  44AC  DB  D0                 10.0 2$:     IN      WMCSTA          ;GET THE WMC STATUS
1725  44AE  E6  01                 7.0          ANI     X.WCLK          ;TRANSLATOR WRITE CLOCK=1
1726  44B0  CA  C5  44             10.0         JZ      3$              ;NO-CONTINUE
1727  44B3  3A  D0  45             13.0         LDA     XLCLK           ;FIRST OCCURANCE OF THE CLOCK?
1728  44B6  A7                    4.0          ANA     A               ;SET CONDITION BITS
1729  44B7  C2  C5  44             10.0         JNZ     3$              ;NO-THEN CONTINUE
1730  44BA  D3  44                 10.0         OUT     TAMT            ;CLEAR THE AMTIE LINES
1731  44BC  3E  20                 7.0          MVI     A,P.LWR         ;CLEAR AMTIE PARITY
1732  44BE  D3  48                 10.0         OUT     PDIAG
1733  44C0  3E  FF                 7.0          MVI     A,@377          ;SET THE TRANSLATOR CLOCK FLAG
1734  44C2  32  D0  45             13.0         STA     XLCLK
1735  44C5  0D                    4.0  3$:     DCR     C               ;DECREMENT THE LOOP COUNT
1736  44C6  C2  8C  44             10.0         JNZ     T2LOP4          ;CONTINUE UNTIL ZERO
1737  44C9  D3  0C                 10.0         OUT     RINST           ;SINGLE STET THE READ PATH
1738  44CB  DB  D0                 10.0         IN      WMCSTA          ;GET THE WRITE MICRO STATUS
1739  44CD  E6  C0                 7.0          ANI     $C0             ;XMC DONE?
1740  44CF  C2  8A  44             10.0         JNZ     T2LOP1          ;NO-CONTINUE
1741  44D2  3E  FF                 7.0          MVI     A,$FF           ;YES-SET THE AMTIES
1742  44D4  D3  44                 10.0         OUT     TAMT            ;TO ONES
1743  44D6  3E  21                 7.0          MVI     A,P.LWR!P.AMTP  ;SET THE PARITY AMTIE
1744  44D8  D3  48                 10.0         OUT     PDIAG           ;TO ONE
1745  44DA  21  30  F8             10.0         LXI     H,-2000         ;LOAD THE WATCHDOG LOOP COUNT
1746  44DD  CD  B9  45             18.0 T2LOP2: CALL    CLKSYS          ;CLOCK THE WRITE PATH
1747  44E0  CD  B9  45             18.0         CALL    CLKSYS          ;CLOCK THE WRITE PATH
1748  44E3  D3  0C                 10.0         OUT     RINST           ;CLOCK THE READ PATH
1749  44E5  DB  01                 10.0         IN      RPCHI           ;GET THE READ CHANNEL STATUS
1750  44E7  E6  40                 7.0          ANI     $40             ;STATUS VALID SET?
1751  44E9  C2  02  45             10.0         JNZ     T2LOP3          ;YES-GET OUT
1752  44EC  0E  01                 7.0          MVI     C,@1            ;SET UP A 16-BIT INCREMENT REGISTER
1753  44EE  06  00                 7.0          MVI     B,@0            ;
1754  44F0  09                    10.0         DAD     B               ;INCREMENT THE LOOP COUNT
1755  44F1  D2  DD  44             10.0         JNC     T2LOP2          ;CONTINUE UNTIL ZERO
```

```
                                                ERR     T2LPA,DUMMX,3
                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 44F4 CD 09 28 18.0                                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)      0004                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 44F7 04                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 44F8 03                                             .BYTE   3
(1) 44F9 CD 15 28 18.0                          DUMMX:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1) 44FC DA 70 43 10.0                                  JC      T2LPA           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>READ CHANNEL ''STATUS VALID'' NOT SET AFTER XMC DONE
                                        ;<FATAL ERROR - TEST ABORTED.
44FF C3 4F 45 10.0                              JMP     EXIT
4502 DB 1A 10.0                         T2LOP3: IN      ECCSTA          ;GET THE ECC STATUS
4504 E6 80 7.0                                  ANI     E.CRC           ;CRC ERROR?
4506 CA 0E 45 10.0                              JZ      T2CN3           ;NO-CONTINUE
4509                                            ERR     T2LPA,T2CN3,3
                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 4509 CD 09 28 18.0                                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)      0005                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 450C 05                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 450D 03                                             .BYTE   3
(1) 450E CD 15 28 18.0                          T2CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1) 4511 DA 70 43 10.0                                  JC      T2LPA           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>CRC ERROR AFTER DATA TRANSFER
4514 DB 1A 10.0                                 IN      ECCSTA          ;GET THE ECC STATUS
4516 E6 10 7.0                                  ANI     E.ACRC          ;ACRC ERROR?
4518 CA 20 45 10.0                              JZ      T2CN4           ;NO-CONTINUE
451B                                            ERR     T2LPA,T2CN4,3
                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
(1) 451B CD 09 28 18.0                                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
(1)      0006                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 451E 06                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 451F 03                                             .BYTE   3
(1) 4520 CD 15 28 18.0                          T2CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
(1) 4523 DA 70 43 10.0                                  JC      T2LPA           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>ACRC ERROR AFTER DATA TRANSFER
4526                                            REQ     @7,0,0,0        ;DUMMY REQUEST TO KEEP THE HOST CPU ALIVE
(1) 4526 CD 06 28 18.0                                  CALL    REQST
(1) 4529 00                                             .BYTE   0               ;DATA PATTERN NUMBER
(1) 452A 00 00                                          .WORD   0               ;SYSTEM ''0'' COUNT
(1) 452C 00 00                                          .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(1) 452E 00                                             .BYTE                   ;DATA COMPARE FLAG IF =1
(1) 452F 07                                             .BYTE   @7              ;REQUEST CODE
4530 3A CD 45 13.0                                      LDA     TPBCNT          ;GET THE BYTE COUNT
4533 3C 4.0                                             INR     A               ;INCREMENT
4534 32 CD 45 13.0                                      STA     TPBCNT          ;SAVE THE NEW BYTE COUNT
4537 A7 4.0                                             ANA     A               ;SET THE CONDITION BITS
4538 C2 70 43 10.0                                      JNZ     T2LPA           ;CONTINUE UNTIL A 256 BYTE RECORD IS WRITTEN
4538                                                    ENDTST  T2LPB
                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
(2) 4538                                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
(2) 4538 CD 06 28 18.0                                  CALL    REQST
(2) 453E 00                                             .BYTE                   ;DATA PATTERN NUMBER
(2) 453F 00 00                                          .WORD                   ;SYSTEM ''0'' COUNT
(2) 4541 00 00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
```

```
(2)  4543  00                                                  .BYTE                   ;DATA COMPARE FLAG IF =1
(2)  4544  07                                                  .BYTE   7               ;REQUEST CODE
(1)  4545  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
(1)  4548  3D               4.0                DCR     A               ;DOWNCOUNT
(1)  4549  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
(1)  454C  F2  6C  43      10.0                JP      T2LPB           ;DO TEST UNTIL TILL = 0
1778
1779 454F  C3  18  28      10.0  EXIT:         JMP     TSTEND
```

```
1781                                    .SBTTL  SUBROUTINE SETDDR
1782  4552                              $
 (1)                                    ; ************************************************************************
1783                                    ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT ''SCLK'' CYCLE.
1784                                    ;       DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
1785                                    ;       'SDDRCC' FOR REFERENCE.
1786  4552                              $
 (1)                                    ; ************************************************************************
1787
1788  4552  F5              12.0  SETDDR: PUSH    PSW             ;SAVE PSW + A
1789  4553  C5              12.0          PUSH    B               ;SAVE B + C
1790  4554  E5              12.0          PUSH    H               ;SAVE H + L
1791  4555  3A  D2  45      13.0          LDA     DPNUM           ;GET THE LAST PATTERN # JSED
1792  4558  3C               4.0          INR     A               ;POINT TO NEXT
1793  4559  32  D2  45      13.0          STA     DPNUM           ;SAVE NUMBER
1794  455C  32  C9  45      13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
1795  455F  AF               4.0          XRA     A               ;
1796  4560  32  CA  45      13.0          STA     SDDRBT          ;SAVE DDR 'B' IMAGE
1797  4563  32  CB  45      13.0          STA     SDDRCT          ;SAVE DDR 'C' IMAGE
1798
1799  4566  3A  C9  45      13.0          LDA     SDDRAT          ;GET DDRA TEMP
1800  4569  E6  3F            7.0          ANI     @77             ;STRIP 2 BITS OFF
1801  456B  07                4.0          RLC
1802  456C  07                4.0          RLC                     ;LEFT JUSTIFY
1803  456D  47                4.0          MOV     B,A             ;TEMP SAVE
1804  456E  3A  CB  45      13.0          LDA     SDDRCT          ;GET DDRC TEMP
1805  4571  E6  03            7.0          ANI     @3              ;SAVE 2 BITS
1806  4573  B0                4.0          ORA     B
1807  4574  32  C6  45      13.0          STA     SDDRA           ;SAVE DDRA DESIRED
1808  4577  D3  D8            10.0         OUT     DDRA            ;LOAD THE REAL DDRA
1809
1810  4579  3A  C9  45      13.0          LDA     SDDRAT          ;GET DDRA AGAIN
1811  457C  E6  C0            7.0          ANI     @300            ;SAVE 2 BITS
1812  457E  07                4.0          RLC
1813  457F  07                4.0          RLC
1814  4580  32  C7  45      13.0          STA     SDDRB           ;SAVE DDRB DATA
1815  4583  D3  D9            10.0         OUT     DDRB            ;LOAD THE REAL DDRB
1816
1817  4585  3A  C8  45      13.0          LDA     SDDRC
1818  4588  D3  DA            10.0         OUT     DDRC
1819  458A  E1                10.0         POP     H               ;RESTORE REGS
1820  458B  C1                10.0         POP     B
1821  458C  F1                10.0         POP     PSW
1822  458D  C9                10.0         RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
458E                                    SSUB
(1)                                     ; ************************************************************************
(1)                                     ;*SUBROUTINE TITLE
(1)                                     ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
458E                                    SD
(1)                                     ; ************************************************************************
(1)                                     ;*DESCRIPTION
(1)                                     ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
458E                                    SP
(1)                                     ; ************************************************************************
(1)                                     ;*PROCEDURE
(1)                                     ;*---------
                                        ;*BGNSUB
                                        ;*      SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*      CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*      CLEAR PORT SELECT FOR TRANSPORT
                                        ;*      CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*      CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*      CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
458E                                    S
(1)                                     ; ************************************************************************
                                        
458E F5             12.0  CLEAR:  PUSH  PSW             ;SAVE THE SELECTED PORT #
458F C5             12.0          PUSH  B
4590 06 00           7.0          MVI   B,0            ;START TO CLEAR AT PORT #0
4592 DB E0          10.0  CLRLP:  IN    INTSTA         ;GET MB SELECT INFO
4594 E6 80           7.0          ANI   BIT7           ;SAVE ONLY THE MASSBUS SELECT BIT
4596 B0              4.0          ORA   B              ;ADD IN THE SELECTED PORT #
4597 D3 E0          10.0          OUT   MBSEL          ;RESET TO THIS PORT
4599 3E 80           7.0          MVI   A,@200         ;LOAD MTA REGISTER #0 SELECT CODE
459B D3 40          10.0          OUT   TCMD           ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
459D AF              4.0          XRA   A              ;CLEAR TU COMMAND A
459E D3 40          10.0          OUT   TCMD           ;
45A0 3E 81           7.0          MVI   A,@201         ;LOAD MTA REGISTER #1 SELECT CODE
45A2 D3 40          10.0          OUT   TCMD           ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
45A4 3E 00           7.0          MVI   A,SELCLR       ;LOAD TU "CLEAR SELECT" COMMAND
45A6 D3 40          10.0          OUT   TCMD           ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
45A8 AF              4.0          XRA   A
45A9 D3 44          10.0          OUT   TAMT           ;CLEAR AMTIE WORD
45AB D3 48          10.0          OUT   PDIAG          ;CLEAR DIAG CONTROL WORD
45AD D3 4C          10.0          OUT   PENAB          ;CLEAR PORT ENABLE WORD
45AF 04              4.0          INR   B              ;POINT TO THE NEXT PORT TO CLEAR
45B0 78              4.0          MOV   A,B
45B1 FE 04           7.0          CPI   4              ;DONE?
45B3 C2 92 45       10.0          JNZ   CLRLP          ;NO - CLEAR THIS PORT ALSO
45B6 C1             10.0          POP   B              ;RESET B & C
45B7 F1             10.0          POP   PSW            ;ALL DONE
45B8 C9             10.0          RET                  ;EXIT
```

```
                                        .SBTTL  SUBROUTINE      CLKSYS
                                        S
                                        ; ************************************************************************
                                        ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
                                        ;       'CLK' BIT TO A '1' THEN A '0'.
                                        ;
                                        S
                                        ; ************************************************************************

45B9 3A C5 45          13.0 CLKSYS: LDA     CCTLWD          ;GET SOFTWARE CLOCK CONTROL IMAGE
45BC F6 40              7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
45BE D3 F0             10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
45C0 E6 3F              7.0          ANI     @77             ;STRIP OFF CLOCK BIT
45C2 D3 F0             10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
45C4 C9                10.0          RET                     ;EXIT - CLOCK CYCLE COMPLETE
```

```
                                        .SBTTL  PROGRAM VARIABLES

                                        ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE
45C5  00                                CCTLWD: .BYTE   0               ;IMAGE OF CLOCK CONTROL WORD
45C6  00                                SDDRA:  .BYTE   0               ;DDRA DATA ACTUALLY LOADED
45C7  00                                SDDRB:  .BYTE   0               ;DDRB DATA ACTUALLY LOADED
45C8  00                                SDDRC:  .BYTE   0               ;DDRC DATA ACTUALLY LOADED

45C9  00                                SDDRAT: .BYTE   0               ;DDRA DATA FROM PGM TABLE
45CA  00                                SDDRBT: .BYTE   0               ;DDRB DATA FROM PGM TABLE
45CB  00                                SDDRCT: .BYTE   0               ;DDRC DATA FROM PGM TABLE
45CC  00                                IMPADCT: .BYTE  0               ;SOFTWARE COPY OF THE PAD COUNT GIVEN TO THE WMC
45CD  00                                TPBCNT: .BYTE   0               ;TEST PASS BYTE COUNTER
45CE  00                                PDCRC:  .BYTE   0               ;PAD/CRC FLAG IF SET, WMC WILL ISSUE A PAD
45CF  00                                UNITMP: .BYTE   0               ;UNIT MAP
45D0  00                                XLCLK:  .BYTE   0               ;XL WRITE CLOCK FLAG
45D1  00                                CLKON:  .BYTE   0               ;SCLOCK ON
45D2  00                                DPNUM:  .BYTE   0               ;DATA PATTERN NUMBER
      0000                                      .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 0088          CBYTL  = 008A          CCTLWD  45C5          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021           CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025           CH6TIE= 0026          CH7TIE= 0027
CKLOP  = 2815          CLEAR   458F           CLKCTL= 00F0          CLKON   45D1
CLKSYS  45B9           CLOCK   4F26           CLRLP   4592          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7           CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082          C.     = 0001          C.AVAI= 0080
C.DP   - 0008          C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          D      =%0002          DATACT= 00D0          DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB           DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 000B          DONE1  = 0045          DONINT= 0010
DPNUM   45D2           DSAVE   4F9E           DSE    = 0006          DUMMX   44F9  G
DUMMY   439D  G        D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          E      =%0003          ECCBAD= 0042          ECCCOR= 0019
ECCOK  - 0041          ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095
EOTCLR= 0003           ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90          ERRCNT= 00D6
ESAVE   4F9F           EXIT    454F           E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008          E.RPE  = 0040
E.STEC- 0001           E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A
FORMAT  4F25           FOUND   43A6           FWDTST= 0061          GCRID  = 0089
GCRSET- 0002           GOODTM= 0092           H      =%0004          HLSAVE  4FA0
IE     = 0008          IMPADC  45CC           INTSTA= 00E0          ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040           I5.5   - 0010          I6.5   = 0020
I7.5   = 0040          KCALL  = 005F          KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8          KEY1   = 0078
KEY10  = 006D          KEY11  = 006E          KEY12  = 006F          KEY13  = 005C
KEY14  = 005D          KEY15  = 005E          KEY16  = 005F          KEY17  = 003C
KEY18  - 003D          KEY19  - 003E          KEY2   - 0079          KEY20  = 003F
KEY3   = 007A          KEY4   = 007B          KEY5   = 0074          KEY6   = 0075
KEY7   = 0076          KEY8   = 0077          KEY9   = 006C          KINTA  = 006F
KLDAD  - 003D          KN0    = 003C          KN1    = 005C          KN2    = 005D
KN3     005E           KN4    - 006C          KN5    = 006D          KN6    = 006E
```

```
KN7    = 0074          KN8    - 0075          KN9    = 0076          KU2    = 0079
KU3    = 007A          KU8    = 0077          L      =%0005          LBLANK= 000F
LCE    = 0008          LCH    = 000C          LCL    = 000D          LCP    = 000E
LC0    = 0000          LC1    = 0001          LC2    = 0002          LC3    = 0003
LC4    = 0004          LC5    = 0005          LC6    = 0006          LC7    = 0007
LC8    = 0008          LC9    = 0009          LDLEDA= 00CA          LDLEDB= 00CB
LDLEDC= 00CC          LDLEDD= 00CD          LDLEDE= 00CE          LDLEDF= 00CF
LKDIAG= 2800          LKKBD  = 004C          LKKEY  = 0049          LKLWMG= 0058
LKLWMP= 0055          LKLWPG= 0052          LKLWPP= 004F          LKMOD7= 0046
LKOPR  = 0046          LPFLG   4F94          LPNUM   4F92          M      =%0006
MBSEL  = 00E0          MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF
MINUS  = 000A          MM     = 8000          MSE    = 0008          MSGN   = 0006
MTACLR= 0000          MT.ARA= 0020          MT.CPE= 0080          MT.DSE= 0001
MT.FWD= 0040          MT.INH= 0008          MT.LWR= 0004          MT.MOT= 0002
MT.NWT= 0080          MT.PEC= 0040          MT.PSB= 0004          MT.PS0= 0001
MT.PS1= 0002          MT.REV= 0020          MT.WRT= 0010          MT.Z   = 0008
M.ATA  = 0080          M.CAPE= 0020          M.CONT= 0080          M.DEM  = 0020
M.EBL  = 0010          M.EXC  = 0008          M.FAIL= 0008          M.ILR  = 0010
M.INIT= 0010          M.OCC  = 0020          M.ONLI= 0001          M.PE   = 0040
M.PORT= C080          M.RDEN= 0002          M.RDPE= 0008          M.RUN  = 0004
M.SCLK= 0001          M.TRA  = 0040          M.UNIT= 0007          M.WCLK= 0040
M.WCLN= 0080          M.WREN= 0080          M5.5   = 0001          M6.5   = 0002
M7.5   = 0004          NOTCAP= 0088          OKAY   = 00FF          OPFRAM= 4300
OPSTRT= 0058          OPVER  = 0040          PADCNT= 00D5          PADCRC= 0080
PDCRC   45CE          PDIAG  = 0048          PEID   = 008A          PENAB  = 004C
PESET  = 0001          PL     = 00B1          PRDD   = 004C          PRENF  = 009C
PS     = 00B2          PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001
P.BCTC= 0040          P.CMDP= 0020          P.INTE= 0080          P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004          P.RPST= 0002
P.RPOE= 0020          P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010
P.SING= 0080          P.STAT= 0002          P.STPE= 0080          P.TACH= 0008
P.TUPR- 0010          P.WCSP= 0004          P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010          P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008
P.WP3E= 0004          P.5VOK= 0002          QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047          RCHOK  = 0046          RCHTST= 000C          RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          REND   = 0014
REQST  = 2806          RESCHR= 00D1          REVTST= 0064          REWIND= 0004
RFIFOL= 0008          RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001
RILL   = 0012          RINST  = 000C          RMCTST= 0008          RMK2   = 0013
RNOP   - 0000          RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001
RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000
RPF1   - 009D          RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016
RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A
RTIER  - 0030          RTM    = 0005          RUNKI  = 0009          RUPTST= 005E
RWDUNL= 0005          R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040
R.DON  - 0002          R.DRDY= 0010          R.END  = 0010          R.ILL  = 0004
R.JVOK= 0004          R.MK2  = 0008          R.PLOD= 0008          R.PLOO= 0010
R.PLO1= 0020          R.POST= 0020          R.STNM= 0002          R.STOP= 0004
R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040          R.VOK  = 0080
R00H   = 0081          R00L   = 0080          R01H   = 0083          R01L   = 0082
R02H   = 0085          R02L   = 0084          R03H   = 0087          R03L   = 0086
R04H   = 0089          R04L   = 0088          R05H   = 008B          R05L   = 008A
```

```
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SDDRA   45C6          SDDRAT  45C9          SDDRB   45C7
SDDRBT  45CA          SDDRC   45C8          SDDRCT  45CB          SELCLR= 0000
SETATA= 00A1          SETDDR  4552          SID   = 0080          SOD   = 0080
SOE   = 0040          SP    =%0008          SSCLK = 0040          SSTEP = 0005
STACK = 4FFF          STATRM= 4F20          STPCT   4F20          STRSP = 5000
TADR00= 0080          TADR01= 0081          TADR02= 0082          TADR03= 0083
TADR04= 0084          TADR05= 0085          TADR06= 0086          TADR07= 0087
TADR10= 0088          TADR11= 0089          TADR12= 008A          TADR13= 008B
TAMT  = 0044          TASEL = 0080          TCMD  = 0040          TC.FWD= 0040
TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010
TEMP    4F99          TEST1   4300          TMF   = 0099          TMRDY = 0040
TPBCNT  45CD          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TST01X  4305          TST02C  43C7          TST2    4357
TST2L   438B          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PS0J= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDY0= 0040          T.RWD = 0010          T.SCLK= 0002          T01CN1  432B  G
T01CN2  433D  G       T2CN3   450E  G       T2CN4   4520  G       T2LOP1  448A
T2LOP2  44DD          T2LOP3  4502          T2LOP4  448C          T2LPA   4370
T2LPB   436C          UIBG  = 00A1          UNITMP  45CF          VALFC   4F98
VALTB   4F95          VELTST= 005B          WDR.P = 0010          WMCCTL= 00D3
WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3
W.ACRC= 0004          W.CRC = 0008          W.DIAG= 0002          W.DONN= 0040
W.ECC = 0010          W.ENAB= 0080          W.ERR = 0020          W.FMT = 0070
W.GCR = 0010          W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002
W.REV = 0004          W.ROME= 0010          W.RST = 0001          W.SKIP= 000F
W.WRIT= 0008          W.XFER= 0020          X     -%000A          XLCLK   45D0
X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001
X.WCLK= 0001          Y     -%000B          .     - 45D3


 ERRORS DETECTED:  0

 *CCM1.A78/PTP,CCM1-NLIST,PARAM,MACRO,LIST,CCM1
 RUN-TIME: 3 4 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:07 SEQ 1080
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  WMC6 - BYTE ASSEMBLY OF READ PATH DATA
                                .SBTTL  TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
                                :ID     WMC6-WRITE MICRO CONTROLLER PART #6
 4300                           ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WMC BYTE ASSEMBLY FROM READ FORWARD DATA
 4300                           SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
                                ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
                                ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
                                ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
                                ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
                                ;*TO 10-HI DEN COMPAT) IN A READ FORWARD DATA TRANSFER. THE TEST WILL
                                ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
                                ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
                                ;*EXPECTED. THE READ PATH WILL BE FILLED WITH 8 SUBGROUPS WORTH OF DATA
                                ;*REGUARDLESS OF THE AMOUNT OF EXPECTED DATA GROUPS TO BE TESTED (USUALLY
                                ;*4 SUBGROUPS OF DATA WILL BE SUFFICIENT TO TEST A PACKING MODE). THE EXTRA
                                ;*SUBGROUPS OF DATA INJECTED WILL ELIMINATE THE NECESSITY OF CALCULATING
                                ;*THE 'SPECIAL' CHARACTERS FROM THE TAPE IN GCR MODE (I.E. CRC, ACRC, &
                                ;*RESID CHARACTERS). EVERY 4 BYTES OF DATA TO BE INJECTED WILL BE TURNED
                                ;*INTO A 5 BYTE SUBGROUP TO SIMULATE DATA FROM THE 'TAPE'. THE 'ECC'
                                ;*WILL BE CALCULATED FOR EACH 7 BYTE OF DATA INJECTED AND ALSO INJECTED
                                ;*AT THE PROPER TIME INTO THE FIFO'S.
 4300                           SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  REQUEST A TU PORT FROM 'HOST' CPU
                                ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
                                ;*  :  THEN-REPORT OPERATOR ERROR
                                ;*  :  ELSE-PROCEED
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  CLEAR ALL PORTS
                                ;*  :  INIT THE SYSTEM
                                ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
                                ;*  :  INIT COUNTERS + CONTROLS
                                ;*  :  INIT PACKING MODE TO 'IMAGE'
                                ;*  :  LOAD BYTE COUNTER + PAD COUNTER
                                ;*  :  RESTART WRITE MICROCONTROLLER
                                ;*  :  CLOCK WMC THROUGH RESTART CODE
                                ;*  :  START 'GCR' READ DATA XFR
                                ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
                                ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  :  CLOCK THE FIFO'S
                                ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
                                ;*  :  WAIT
```

```
                                ;*   :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*   :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
                                ;*   :  SETUP A WATCHDOG TIMER
                                ;*   :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
                                ;*   :  IF WATCHDOG TIMER=0
                                ;*   :  :  THEN-REPORT NO 'SCLK'
                                ;*   :  :  ELSE-CONT
                                ;*   :  ENDIF
                                ;*   :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
                                ;*   :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
                                ;*   :  IF 18 BIT DATA SAME AS EXPECTED
                                ;*   :  :  THEN-PROCEED
                                ;*   :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
                                ;*   :  ENDIF
                                ;*   :  POINT TO NEXT SET OF EXPECTED DATA
                                ;*   :  IF ALL DONE WITH CURRENT XFR
                                ;*   :  :  THEN-UPDATE TO NEXT PACKING MODE
                                ;*   :  .     IF ALL MODES DONE
                                ;*   :  :     :  THEN-EXIT
                                ;*   :  :     :  ELSE-CONTINUE
                                ;*   :  :     ENDIF
                                ;*   :  :  ELSE-CONTINUE
                                ;*   :  ENDIF
                                ;*   :  DO UNTIL ALL 'SCLK' CHECKED FOR SELECTED XFR
                                ;*  ENDDO
                                ;*ENDTST
4300                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC6 MICRO TEST 01
                                ;*WMC6 MICRO ERROR 01
                                ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                ;*M8959 (WMC) & ALL OTHER BOARDS
                                ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*WMC6 MICRO TEST 01
                                ;*WMC6 MICRO ERROR 02
                                ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                ;*M8959 (WMC) & ALL OTHER BOARDS
                                ;*TIMEOUT WHILE WAITING FOR 'SCLK TO SET IN DBUSSTA
                                ;*AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
                                ;*
                                ;*WMC6 MICRO TEST 01
                                ;*WMC6 MICRO ERROR 03
                                ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                ;*M8959 (WMC) & ALL OTHER BOARDS
                                ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
                                ;*DURING A READ FORWARD COMMAND
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*WMC6 MICRO TEST 01
                                ;*WMC6 MICRO ERROR 04
```

```
1425                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1426                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1427                                    ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
1428                                    ;*ACTUAL = NNNN
1429                                    ;*EXPECTED = NNNN
1430                                    ;*
1431                                    ;*WMC6 MICRO TEST 01
1432                                    ;*WMC6 MICRO ERROR 05
1433                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1434                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1435                                    ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
1436                                    ;*ACTUAL = NNNN
1437                                    ;*EXPECTED   NNNN
1438                                    ;*
1439                                    ;*WMC6 MICRO TEST 01
1440                                    ;*WMC6 MICRO ERROR 06
1441                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1442                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1443                                    ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
1444                                    ;*ACTUAL   NNNN
1445                                    ;*EXPECTED = NNNN
1446  4300                              S
 (1)                                    ; ************************************************************************************
1447
1448  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01              7.0                     MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1449                                    ;%WMC BYTE ASSY FOR READ FORWARD XFR
1450                                    ;&M8959 (WMC) & ALL OTHER BOARDS
1451                                    ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
1452
1453  4305                                      REQ     7
 (1)  4305  CD  06  28         18.0                     CALL    REQST
 (1)  4308  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (1)  4309  00  00                                      .WORD                   ;SYSTEM '*' COUNT
 (1)  430B  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                          .BYTE   7               ;REQUEST CODE
1454  430F                                RIN     R12L                    ;GET THE SELECTED PORT FROM 'HOST'
 (1)  430F  DB  94             10.0                     IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                  4.0                     MOV     A,A             ;RETRY LINK
1455  4312  32  2D  4A          13.0          STA       UNITMP                  ;SAVE IT
1456  4315  A7                   4.0          ANA       A                       ;SET THE CONDITION BITS
1457  4316  C2  27  43          10.0          JNZ       FOUND                   ;SEE IF ANY WERE SELECTED IN 'HOST'
1458  4319                                    ERR       EXIT,NOUNIT
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28         18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                         .BYTE
 (1)  431E  CD  15  28         18.0                     CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  6F  45         10.0                     JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1459                                    ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
1460                                    ;>FATAL ERROR - TEST ABORTED
```

```
1461                                        ;*REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
1462  4324  C3  6F  45        10.0                JMP     EXIT
1463                                        ;HERE IF THE HOST CPU INDICATED A PORT TO USE
1464
1465  4327  CD  7C  49        18.0  FOUND:  CALL    CLEAR           ;INIT ALL THE PORTS IN THE PORT BOARD
1466  432A  3A  2D  4A        13.0          LDA     UNITMP          ;GET THE RESPONSE FROM 'HOST' CPU
1467  432D  06  00             7.0          MVI     B,0             ;INIT 'B'
1468  432F  E6  01             7.0          ANI     1               ;IS UNIT 0 SELECTED?
1469  4331  C2  47  43        10.0          JNZ     FPORT           ;YUP - FOUND THE PORT TO USE
1470  4334  04                 4.0          INR     B               ;NO - COUNT THE PORT
1471  4335  3A  2D  4A        13.0          LDA     UNITMP          ;GET RESPONSE AGAIN
1472  4338  E6  02             7.0          ANI     2               ;PORT 1 SELECTED?
1473  433A  C2  47  43        10.0          JNZ     FPORT           ;YUP
1474  433D  04                 4.0          INR     B               ;COUNT THE PORT
1475  433E  3A  2D  4A        13.0          LDA     UNITMP          ;GET RESPONSE
1476  4341  E6  04             7.0          ANI     4               ;PORT 2 SELECTED?
1477  4343  C2  47  43        10.0          JNZ     FPORT           ;YUP
1478  4346  04                 4.0          INR     B               ;NO - ASSUME PORT 3
1479
1480  4347  78                 4.0  FPORT:  MOV     A,B             ;PUT UNIT # IN AC
1481  4348  32  2E  4A        13.0          STA     SUNIT           ;SAVE THE UNIT UNDER TEST
1482  434B  DB  E0             10.0          IN      INTSTA          ;GET THE PORT SELECT STATUS
1483  434D  E6  80             7.0          ANI     BIT7
1484  434F  B0                 4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1485  4350  D3  E0             10.0          OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
1486  4352  3E  60             7.0          MVI     A,P.LWR!P.LCS   ;GET THE LOOP-COMMAND-STATUS BIT
1487  4354  D3  48             10.0          OUT     PDIAG           ;SELECT IT
1488  4356  AF                 4.0          XRA     A
1489  4357  D3  44             10.0          OUT     TAMT            ;CLEAR THE AMTIE LINES
1490  4359  D3  D2             10.0          OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS
1491  435B  D3  D2             10.0          OUT     TRKENA
1492  435D  3E  05             7.0          MVI     A,5
1493  435F  32  2C  4A        13.0          STA     FMTNUM          ;INIT THE PACKING MODE TO IMAGE MODE
1494  4362  3E  04             7.0          MVI     A,P.RPEN        ;SET READ PATH ENABLE
1495  4364  D3  4C             10.0          OUT     PENAB           ;NOW
1496  4366  3E  10             7.0          MVI     A,RDCLK         ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1497  4368  D3  F0             10.0          OUT     CLKCTL
1498  436A  AF                 4.0          XRA     A               ;SETUP TO READ FWD 1ST
1499  436B  32  2F  4A        13.0          STA     REVERSE         ;=0 FOR READ FWD,  -1 FOR READ REV
1500  436E  D3  C0             10.0          OUT     DBUSCTL         ;CLEAR THE DATA BUS CONTROL WORD
1501                                        ;INIT THE COUNTER CONTROL AND COUNTERS
1502
1503  4370                                  T1LOOP: INICNT
 (1)  4370  3E  30             7.0                          MVI     A,@060          ;INIT THE
 (1)  4372  D3  D7             10.0                         OUT     CNTCTL          ;BYTE COUNTER
 (1)  4374  3E  70             7.0                          MVI     A,@160          ;INIT THE
 (1)  4376  D3  D7             10.0                         OUT     CNTCTL          ;PAD COUNTER
 (1)  4378  3E  B0             7.0                          MVI     A,@260          ;INIT THE
 (1)  437A  D3  D7             10.0                         OUT     CNTCTL          ;ERROR COUNTER
1504  437C                                        CLRECT
 (1)  437C  AF                 4.0                          XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  437D  D3  D6             10.0                         OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  437F  D3  D6             10.0                         OUT     ERRCNT          ;CLEAR BITS 15-8
1505  4381  3A  2C  4A        13.0          LDA     FMTNUM          ;GET THE SELECTED PACKING MODE #
```

```
1506  4384  87                    4.0          ADD     A               ;DOUBLE THE COUNT - 2 ADDR PER ENTRY IN TABLE
1507  4385  5F                    4.0          MOV     E,A
1508  4386  16  00                7.0          MVI     D,0
1509  4388  21  AF  48           10.0  T1RF:   LXI     H,EDFWD         ;POINT TO READ FORWARD
1510  438B  19                   10.0          DAD     D               ;GET OFFSET TO EXPECTED DATA
1511  438C  7E                    7.0          MOV     A,M             ;GET LOW BYTE OF ADDRESS
1512  438D  32  D2  48           13.0          STA     EXPADR          ;SAVE IT
1513  4390  23                    6.0          INX     H               ;POINT TO THE HIGH BYTE
1514  4391  7E                    7.0          MOV     A,M             ;GET IT
1515  4392  32  D3  48           13.0          STA     EXPADR+1        ;SAVE HIGH BYTE OF EXPECTED DATA ADDRESS
1516  4395  3A  2C  4A           13.0          LDA     FMTNUM          ;GET THE PACKING MODE # AGAIN
1517  4398  5F                    4.0          MOV     E,A
1518  4399  16  00                7.0          MVI     D,0
1519  439B  21  CB  48           10.0          LXI     H,ESCLKT        ;POINT TO TABLE OF EXPECTED SCLKS/MODE
1520  439E  19                   10.0          DAD     D               ;POINT TO TABLE OFFSET #
1521  439F  7E                    7.0          MOV     A,M             ;GET THE EXPECTED # SCLKS/MODE SELECTED
1522  43A0  32  2B  4A           13.0          STA     ESCLK           ;SAVE THE NUMBER FOR THIS PASS
1523  43A3  3A  2C  4A           13.0          LDA     FMTNUM          ;GET THE PACKING MODE # AGAIN
1524  43A6  07                    4.0          RLC
1525  43A7  07                    4.0          RLC
1526  43A8  07                    4.0          RLC
1527  43A9  07                    4.0          RLC
1528  43AA  E6  70                7.0          ANI     $70             ;POSITION THE FMT #
1529  43AC  D3  D0               10.0          OUT     DATACTL         ;LOAD THE DATA CONTROL WORD
1530  43AE                                     ROUT    R02H            ;SAVE FOR ERROR PRINTING
 (1)  43AE  D3  85               10.0                  OUT     R02H            ;WRITE AC INTO R02H
 (1)  43B0  7F                    4.0                  MOV     A,A             ;RETRY LINK
1531  43B1  3E  9B                7.0          MVI     A,@233          ;SET THE DDR CONTROL TO 'IN'
1532  43B3  D3  DB               10.0          OUT     DDRCTL          ;TO READ THE 'DDR'
1533  43B5  3E  00                7.0          MVI     A,@0            ;GET THE RESIDUAL CHAR WORD
1534  43B7  D3  D1               10.0          OUT     RESCHR          ;LOAD IT
1535  43B9  3E  32                7.0          MVI     A,50            ;SETUP THE BYTE COUNTER FOR 8 DATA GROUPS
1536  43BB  D3  D4               10.0          OUT     BYTCNT
1537  43BD  AF                    4.0          XRA     A
1538  43BE  D3  D4               10.0          OUT     BYTCNT
1539  43C0  3E  06                7.0          MVI     A,6             ;NEED 6 PAD CHARACTERS FOR THE XFR
1540  43C2  D3  D5               10.0          OUT     PADCNT          ;LOAD IT
1541  43C4  AF                    4.0          XRA     A
1542  43C5  D3  D5               10.0          OUT     PADCNT
1543  43C7                                     ROUT    R02L            ;CLEAR OUT THE REST OF REG 2 IN CAS
 (1)  43C7  D3  84               10.0                  OUT     R02L            ;WRITE AC INTO R02L
 (1)  43C9  7F                    4.0                  MOV     A,A             ;RETRY LINK
1544  43CA                                     ROUT    R05L            ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
 (1)  43CA  D3  8A               10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  43CC  7F                    4.0                  MOV     A,A             ;RETRY LINK
1545  43CD  32  2A  4A           13.0          STA     SCLKCT          ;INIT THE SCLK DETECTED COUNTER
1546
1547                                  ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1548
1549  43D0  3E  01                7.0          MVI     A,W.RST         ;GET THE WMC RESTART BIT
1550  43D2  D3  D3               10.0          OUT     WMCCTL          ;SET RESTART
1551  43D4  3E  15                7.0          MVI     A,SSTEP!RDCLK   ;GET SINGLE STEP FOR WMC
1552  43D6  D3  F0               10.0          OUT     CLKCTL          ;SET SINGLE STEP MODE
1553  43D8  32  30  4A           13.0          STA     CCTLWD          ;SAVE THE WORD
```

```
1554  43DB  CD  81  4A    18.0          CALL   CLKSYS          ;CLOCK THE SYSTEM TO CAUSE RESTART TO
1555  43DE  CD  81  4A    18.0          CALL   CLKSYS          ;TO TAKE HOLD
1556  43E1  AF           4.0           XRA    A
1557  43E2  D3  D3        10.0          OUT    WMCCTL          ;FINISH THE RESTART CYCLE
1558
1559                                 ;CLOCK THE WMC THRU ITS RESTART AREA
1560
1561  43E4  06  B4        7.0           MVI    B,180           ;GET A LOOP COUNT
1562  43E6  CD  81  4A    18.0    CLKRS: CALL   CLKSYS          ;CLOCK THE SYSTEM
1563  43E9  05           4.0           DCR    B               ;DECREMENT THE LOOP COUNTER
1564  43EA  C2  E6  43    10.0          JNZ    CLKRS           ;CLOCK TILL COUNT RUN OUT
1565
1566  43ED  3E  90        7.0           MVI    A,W.GCR.W.ENAB  ;SETUP FOR A GCR DATA XFR
1567  43EF  D3  D3        10.0    T1ST:  OUT    WMCCTL          ;LOAD THE WORD IN WMC
1568
1569                                 ;CALCULATE THE 1ST DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1570
1571  43F1  21  8F  48    10.0    C1EC1: LXI    H,DG1           ;POINT THE THE 1ST DATA GROUP DATA TABLE
1572  43F4  06  07        7.0           MVI    B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1573  43F6  CD  42  4A    18.0          CALL   CLECC           ;INIT THE ECC CHARACTER
1574
1575  43F9  7E            7.0    C1EC1L: MOV    A,M             ;GET A DATA BYTE FROM THE TABLE
1576  43FA  CD  47  4A    18.0          CALL   ECC             ;CALC ECC WITH THIS CHARACTER
1577  43FD  23           6.0           INX    H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1578  43FE  05           4.0           DCR    B               ;DECREMENT THE COUNTER OF DATA BYTES
1579  43FF  C2  F9  43    10.0          JNZ    C1EC1L          ;STAY HERE TILL ALL DATA BYTES USED
1580  4402  3A  6E  4A    13.0          LDA    ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1581  4405  32  96  48    13.0          STA    DG1ECC          ;SAVE THE CALCULATED ECC CHARACTER
1582
1583                                 ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1584
1585  4408  21  97  48    10.0    C1EC2: LXI    H,DG2           ;POINT THE THE 2ND DATA GROUP DATA TABLE
1586  4408  06  07        7.0           MVI    B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1587  440D  CD  42  4A    18.0          CALL   CLECC           ;INIT THE ECC CHARACTER
1588
1589  4410  7E            7.0    C1EC2L: MOV    A,M             ;GET A DATA BYTE FROM THE TABLE
1590  4411  CD  47  4A    18.0          CALL   ECC             ;CALC ECC WITH THIS CHARACTER
1591  4414  23           6.0           INX    H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1592  4415  05           4.0           DCR    B               ;DECREMENT THE COUNTER OF DATA BYTES
1593  4416  C2  10  44    10.0          JNZ    C1EC2L          ;STAY HERE TILL ALL DATA BYTES USED
1594  4419  3A  6E  4A    13.0          LDA    ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1595  441C  32  9E  48    13.0          STA    DG2ECC          ;SAVE THE CALCULATED ECC CHARACTER
1596  441F  3E  A8        7.0    T1PLO: MVI    A,R.PLO1!R.TBJN!R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1597                                                                ;SET TIE BUS JAM AND PLO DISABLE
1598  4421  D3  09        10.0          OUT    RPCTL
1599  4423  AF           4.0           XRA    A
1600  4424  D3  0A        10.0          OUT    RTIEB           ;CLEAR THE TIE BUS
1601  4426  D3  08        10.0          OUT    RFIFOL          ;CLOCK THE FIFO'S
1602  4428  3E  0D        7.0           MVI    A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1603  442A  D3  0B        10.0          OUT    RCMD
1604
1605  442C  00           4.0           NOP                    ;WAIT
1606  442D  00           4.0           NOP                    ;WAIT
1607  442E  00           4.0           NOP
```

```
1608  442F  00              4.0          NOP
1609  4430  00              4.0          NOP
1610
1611  4431  3E A9           7.0          MVI     A,R.PLO1!R.STPC.R.PLOD.R.TBJN   ;STOP THE READ PATH
1612                                                                            ;SET TIE BUS JAM AND PLO DISABLE
1613  4433  D3 09          10.0          OUT     RPCTL
1614
1615  4435  0E 03           7.0          MVI     C,@3                    ;INIT THE LOOP COUNT FOR 6 DATA GROUPS
1616
1617  4437  21 8F 48       10.0  T1SG1:  LXI     H,DG1                   ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
1618  443A  CD 43 48       18.0          CALL    CONVRT                  ;CONVERT SUBGROUP - 4X5
1619  443D  CD 58 48       18.0          CALL    LOADIT                  ;INJECT THE DATA INTO THE FIFO
1620
1621  4440  21 93 48       10.0  T1SG2:  LXI     H,SG2                   ;POINT TO THE 2ND SUBGROUP OF DATA
1622  4443  CD 43 48       18.0          CALL    CONVRT                  ;MAKE IT 4X5
1623  4446  CD 58 48       18.0          CALL    LOADIT                  ;INJECT IT INTO FIFO
1624
1625  4449  21 97 48       10.0  T1SG3:  LXI     H,DG2                   ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
1626  444C  CD 43 48       18.0          CALL    CONVRT                  ;MAKE IT 4X5
1627  444F  CD 58 48       18.0          CALL    LOADIT                  ;INJECT THE DATA INTO FIFO
1628
1629  4452  21 9B 48       10.0  T1SG4:  LXI     H,SG4                   ;POINT TO THE 4TH SUBGROUP OF DATA
1630  4455  CD 43 48       18.0          CALL    CONVRT                  ;MAKE IT 4X5
1631  4458  CD 58 48       18.0          CALL    LOADIT                  ;INJECT THE DATA INTO FIFO
1632
1633  445B  0D              4.0          DCR     C                       ;DECREMENT THE LOOP COUNTER
1634  445C  C2 37 44       10.0          JNZ     T1SG1                   ;KEEP INJECTING DATA TILL 6 GROUPS IN FIFO
1635
1636  445F  3E 0B           7.0          MVI     A,DIARD                 ;LOAD THE DIAGNOSTIC READ
1637  4461  D3 0B          10.0          OUT     RCMD                    ;COMMAND
1638  4463  11 01 00       10.0  T1CLKA: LXI     D,1                     ;SET WATCH DOG INCREMENT
1639  4466  21 68 C5       10.0          LXI     H,-15000                ;SET WATCH DOG COUNT TO 15000
1640  4469  D3 0C          10.0  T1CLK:  OUT     RINST                   ;STEP THE READ PATH
1641  446B  CD 81 4A       18.0          CALL    CLKSYS                  ;CLOCK THE WMC
1642  446E  DB C0          10.0          IN      DBUSSTA                 ;GET THE DATABUS STATUS
1643  4470  E6 02           7.0          ANI     T.SCLK                  ;SEE IF SCLK UP - DATA READY
1644  4472  C2 87 44       10.0          JNZ     T1SCLK                  ;JUMP IF SCLK UP
1645  4475  19              10.0          DAD     D
1646  4476  D2 69 44       10.0          JNC     T1CLK                   ;STAY IN LOOP UNLESS TIMEOUT
1647
```

```
1649  4479                                          ERR     T1LOOP,DUMM1,3
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4479  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447C  02                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447D  03                                              .BYTE   3
 (1)  447E  CD  15  28        18.0                  DUMM1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4481  DA  70  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1650                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
1651                                    ;>AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
1652                                    ;<FATAL ERROR - TEST ABORTED
1653  4484  C3  6F  45        10.0            JMP     EXIT
1654
1655                                    ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO  0
1656                                    ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
1657                                    ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDICATING THE DATA IS
1658                                    ;ON THE MASSBUS
1659
1660  4487  06  FF          7.0   T1SCLK: MVI     B,$FF           ;SETUP A DROP DEAD COUNT
1661  4489  D3  0C         10.0   T1SCKL: OUT     RINST           ;CLOCK THE READ PATH ONCE
1662  448B  CD  81  4A      18.0           CALL    CLKSYS          ;CLOCK THE WMC TO CLEAR 'SCLK'
1663  448E  DB  C0         10.0           IN      DBUSSTA         ;SEE IF 'SCLK' IS GONE
1664  4490  E6  02          7.0           ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
1665  4492  CA  A7  44      10.0           JZ      T1SCK0          ;JUMP IF 'SCLK' = 0
1666  4495  05                4.0           DCR     B               ;DECREMENT THE COUNTER
1667  4496  C2  89  44      10.0           JNZ     T1SCKL          ;LOOP BACK TILL COUNTER -0
1668  4499                                  ERR     T1LOOP,DUMM2,3
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4499  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0003                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  449C  03                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  449D  03                                              .BYTE   3
 (1)  449E  CD  15  28        18.0                  DUMM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A1  DA  70  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1669                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1670                                    ;>DURING A READ FORWARD COMMAND
1671                                    ;<FATAL ERROR - TEST ABORTED
1672  44A4  C3  6F  45        10.0            JMP     EXIT
1673                                    ;HERE WHEN 'SCLK' WAS SET - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
1674                                    ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
1675
1676  44A7  DB  D8         10.0   T1SCK0: IN      DDRA            ;GET THE DDR A DATA
1677  44A9  32  27  4A     13.0           STA     SDDRA           ;SAVE IT
1678  44AC                                ROUT    R16L            ;SAVE FOR PRINTING
 (1)  44AC  D3  9C         10.0                   OUT     R16L            ;WRITE AC INTO R16L
 (1)  44AE  7F              4.0                   MOV     A,A             ;RETRY LINK
1679  44AF  DB  D9         10.0           IN      DDRB            ;GET DDR B DATA
1680  44B1  32  28  4A     13.0           STA     SDDRB           ;SAVE IT
1681  44B4                                ROUT    R16H            ;SAVE FOR ERROR REPORTING
 (1)  44B4  D3  9D         10.0                   OUT     R16H            ;WRITE AC INTO R16H
 (1)  44B6  7F              4.0                   MOV     A,A             ;RETRY LINK
1682  44B7  DB  DA         10.0           IN      DDRC            ;GET DDR C DATA
1683  44B9  E6  03          7.0           ANI     3
1684  44BB  32  29  4A     13.0           STA     SDDRC           ;SAVE IT
```

```
                                                ORI     BIT7            ;ADD IN THE PRINT FLAG
                                                ROUT    R17L            ;SAVE IT FOR PRINTING
 (1)  44C0  D3  9E              10.0                    OUT     R17L            ;WRITE AC INTO R17L
 (1)  44C2  7F                   4.0                    MOV     A,A             ;RETRY LINK

                                LDA     SCLKCT          ;GET THE CURRENT SCLK COUNT
                                INR     A               ;COUNT THIS SCLK
                                STA     SCLKCT          ;SAVE THE NEW COUNT
                                ROUT    R05L            ;ALSO IN CAS FOR ERROR REPORTING
 (1)  44CA  D3  8A              10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  44CC  7F                   4.0                    MOV     A,A             ;RETRY LINK
                                XRA     A
                                ROUT    R05H
 (1)  44CE  D3  8B              10.0                    OUT     R05H            ;WRITE AC INTO R05H
 (1)  44D0  7F                   4.0                    MOV     A,A             ;RETRY LINK

                        ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED

                        T1CKA:  LHLD    EXPADR          ;POINT TO THE TABLE OF EXPECTED DATA
                                MOV     A,M             ;GET BITS 7-0
                                ROUT    R14L            ;SAVE FOR ERROR REPORTING
 (1)  44D5  D3  98              10.0                    OUT     R14L            ;WRITE AC INTO R14L
 (1)  44D7  7F                   4.0                    MOV     A,A             ;RETRY LINK
                                INX     H               ;POINT TO NEXT BYTE
                                MOV     A,M             ;GET BITS 15-8
                                ROUT    R14H            ;SAVE IT
 (1)  44DA  D3  99              10.0                    OUT     R14H            ;WRITE AC INTO R14H
 (1)  44DC  7F                   4.0                    MOV     A,A             ;RETRY LINK
                        INX     H
                        MOV     A,M             ;GET BITS 17-16
                        ORI     BIT7            ;ADD IN THE PRINT FLAG
                        ROUT    R15L            ;SAVE FOR ERROR REPORT
 (1)  44E1  D3  9A              10.0                    OUT     R15L            ;WRITE AC INTO R15L
 (1)  44E3  7F                   4.0                    MOV     A,A             ;RETRY LINK
                        LDA     SDDRA           ;GET THE DDR A DATA
                        ROUT    ADATA           ;SAVE IN CASE OF ERROR DETECTED
 (1)  44E7  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44E9  7F                   4.0                    MOV     A,A             ;RETRY LINK
                        LHLD    EXPADR          ;POINT TO THE EXPECTED DATA TABLE
                        CMP     M               ;ACTUAL DATA = EXPECTED?
                        JZ      T1CKAC          ;JUMP IF OK
                        MOV     A,M             ;GET THE EXPECTED DATA
                        ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  44F2  D3  95              10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44F4  7F                   4.0                    MOV     A,A             ;RETRY LINK
                        ERRB    T1LOOP,T1CKAC,@10
 (1)                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44F5  CD  12  28          18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F8  04                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F9  08                                          .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  44FA  CD  15  28          18.0            T1CKAC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FD  DA  70  43          10.0                    JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                        ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
```

```
1716  4500  23                    6.0  T1CKB:  INX     H                ;POINT TO THE EXPECTED DATA FOR DDR B
1717  4501  3A  28  4A            13.0         LDA     SDDRB            ;GET THE DDR B DATA
1718  4504                               ROUT    ADATA                ;SAVE FOR ERROR REPORT
 (1)  4504  D3  94                10.0                 OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4506  7F                     4.0                 MOV     A,A              ;RETRY LINK
1719  4507  BE                    7.0         CMP     M                ;ACTUAL = EXPECTED DATA?
1720  4508  CA  14  45            10.0         JZ      T1CKBC           ;JUMP IF OK
1721  450B  7E                     7.0         MOV     A,M              ;GET THE EXPECTED DATA
1722  450C                               ROUT    EDATA                ;SAVE FOR ERROR REPORT
 (1)  450C  D3  95                10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  450E  7F                     4.0                 MOV     A,A              ;RETRY LINK
1723  450F                               ERRB    T1LOOP,T1CKBC,@10
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  450F  CD  12  28            18.0                 CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0005                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4512  05                                         .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4513  08                                         .BYTE   @10              ;PRINT ROUTINE NUMBER
 (1)  4514  CD  15  28            18.0         T1CKBC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4517  DA  70  43            10.0                 JC      T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1724                           ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
1725
1726  451A  23                    6.0  T1CKC:  INX     H                ;POINT TO EXPECTED DDR C DATA
1727  451B  3A  29  4A            13.0         LDA     SDDRC            ;GET DDR C DATA
1728  451E                               ROUT    ADATA                ;SAVE FOR ERROR REPORT
 (1)  451E  D3  94                10.0                 OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4520  7F                     4.0                 MOV     A,A              ;RETRY LINK
1729  4521  BE                    7.0         CMP     M                ;ACTUAL DATA = EXPECTED DDRC DATA?
1730  4522  CA  2E  45            10.0         JZ      T1CKCC           ;JUMP IF OK
1731  4525  7E                     7.0         MOV     A,M              ;GET THE EXPECTED DATA
1732  4526                               ROUT    EDATA                ;SAVE FOR ERROR REPORT
 (1)  4526  D3  95                10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  4528  7F                     4.0                 MOV     A,A              ;RETRY LINK
1733  4529                               ERRB    T1LOOP,T1CKCC,@10
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4529  CD  12  28            18.0                 CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0006                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  452C  06                                         .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  452D  08                                         .BYTE   @10              ;PRINT ROUTINE NUMBER
 (1)  452E  CD  15  28            18.0         T1CKCC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4531  DA  70  43            10.0                 JC      T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1734                           ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
1735
1736  4534  2A  D2  48            16.0         LHLD    EXPADR           ;GET THE CURRENT EXPECTED DATA ADDR.
1737  4537  23                     6.0         INX     H
1738  4538  23                     6.0         INX     H                ;UPDATE THE EXPECTED ADDRESS POINTER
1739  4539  23                     6.0         INX     H                ;TO THE DATA
1740  453A  22  D2  48            16.0         SHLD    EXPADR           ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
1741  453D  3A  2A  4A            13.0         LDA     SCLKCT           ;GET THE SCLK COUNT
1742  4540  21  2B  4A            10.0         LXI     H,ESCLK          ;POINT TO THE EXPECTED SCLK COUNT
1743  4543  BE                    7.0         CMP     M                ;DONE WITH THIS TEST PASS?
1744  4544  C2  63  44            10.0         JNZ     T1CLKA           ;JUMP IF STILL EXPECTING MORE SCLKS
1745
1746                           ;UPDATE THE PACKING MODE NUMBER FOR THE NEXT PASS
1747
```

```
1748  4547  3A  2C  4A      13.0          LDA     FMTNUM          ;GET THE CURRENT NUMBER
1749  454A  3C                 4.0          INR     A
1750  454B  32  2C  4A      13.0          STA     FMTNUM          ;SAVE THE NEW NUMBER
1751  454E  FE  05               7.0          CPI     5               ;READY TO DO MODE #6??
1752  4550  C2  5B  45        10.0          JNZ     T1PM6           ;SEE IF TIME TO DO MODE #0
1753  4553  3E  06               7.0          MVI     A,6             ;YUP (ALREADY DID #5 ON 1ST PASS)
1754  4555  32  2C  4A      13.0          STA     FMTNUM          ;SAVE IT
1755  4558  C3  70  43        10.0          JMP     T1LOOP          ;DO MODE 6
1756  455B  FE  06               7.0  T1PM6:  CPI     6               ;READY TO START AT MODF 0
1757  455D  CA  68  45        10.0          JZ      T1PM0           ;YUP
1758  4560  FE  07               7.0          CPI     7               ;DONE WITH ALL THE PACKING MODES?
1759  4562  CA  6F  45        10.0          JZ      EXIT            ;YUP
1760  4565  C3  70  43        10.0          JMP     T1LOOP          ;DO THE NEXT PACKING MODE
1761
1762  4568  AF                 4.0  T1PM0:  XRA     A               ;START AT MODE 0
1763  4569  32  2C  4A      13.0          STA     FMTNUM          ;SAVE THE PACKING MODE #
1764  456C  C3  70  43        10.0          JMP     T1LOOP          ;DO THE MODE
1765
1766  456F                               EXIT:   ENDTST  FOUND
  (1)                              ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  456F                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  456F  CD  06  28      18.0                                 CALL    REQST
  (2)  4572  00                                                   .BYTE                   ;DATA PATTERN NUMBER
  (2)  4573  00  00                                               .WORD                   ;SYSTEM "" COUNT
  (2)  4575  00  00                                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4577  00                                                   .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4578  07                                                   .BYTE   7               ;REQUEST CODE
  (1)  4579  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  457C  3D                 4.0          DCR     A               ;DOWNCOUNT
  (1)  457D  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
  (1)  4580  F2  27  43        10.0          JP      FOUND           ;DO TEST UNTIL TILL - 0
1767  4583  CD  7C  49        18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
```

```
                                .SBTTL  TEST 2 - WMC BYTE ASSEMBLY - READ REVERSE
                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WMC BYTE ASSEMBLY FROM READ REVERSE DATA
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
                                ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
                                ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
                                ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
                                ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
                                ;*TO 10-HI DEN COMPAT) IN A READ REVERSE DATA TRANSFER. THE TEST WILL
                                ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
                                ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
                                ;*EXPECTED. THE READ PATH WILL BE FILLED WITH 8 SUBGROUPS WORTH OF DATA
                                ;*REGUARDLESS OF THE AMOUNT OF EXPECTED DATA GROUPS TO BE TESTED (USUALLY
                                ;*4 SUBGROUPS OF DATA WILL BE SUFFICIENT TO TEST A PACKING MODE). THE
                                ;*FIRST TWO GROUPS OF DATA ARE SIMULATED 'CRC' DATA GROUP AND THE 'RESID'
                                ;*DATA GROUP. THE RESIDUAL DATA GROUP WILL CONTAIN NO DATA CHARACTERS,
                                ;*JUST THE 6 PAD CHARACTERS AND A FAKE ACRC & FAKE ECC CHARACTER (ALL 0'S).
                                ;*THE CRC GROUP WILL CONTAIN A RESIDUAL CHARACTER THAT INDICATED NO DATA
                                ;*BYTES IN THE RESIDUAL DATA GROUP. EVERY 4 CHARACTER BYTES INJECTED WILL
                                ;*BE TURNED INTO A 5 BYTE SUBGROUP TO SIMULATE DATA FROM THE 'TAPE'. THE 'ECC'
                                ;*WILL BE CALCULATED FOR EACH 7 BYTE OF DATA INJECTED AND ALSO INJECTED
                                ;*AT THE PROPER TIME INTO THE FIFO'S. ALL DATA INJECTED INTO THE FIFO'S
                                ;*WILL BE REVERSED TO SIMULATE A READ REVERSE DATA TRANSFER.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  REQUEST A TU PORT FROM 'HOST' CPU
                                ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
                                ;*  :  THEN-REPORT OPERATOR ERROR
                                ;*  :  ELSE-PROCEED
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  CLEAR ALL PORTS
                                ;*  :  INIT THE SYSTEM
                                ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
                                ;*  :  INIT COUNTERS + CONTROLS
                                ;*  :  INIT PACKING MODE TO 'IMAGE'
                                ;*  :  LOAD BYTE COUNTER + PAD COUNTER
                                ;*  :  RESTART WRITE MICROCONTROLLER
                                ;*  :  CLOCK WMC THROUGH RESTART CODE
                                ;*  :  START 'GCR' READ DATA XFR
                                ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
                                ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  :  CLOCK THE FIFO'S
                                ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
```

```
                                ;*   :  WAIT
                                ;*   :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*   :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
                                ;*   :  SETUP A WATCHDOG TIMER
                                ;*   :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
                                ;*   :  IF WATCHDOG TIMER=0
                                ;*   :  :  THEN-REPORT NO 'SCLK'
                                ;*   :  :  ELSE-CONT
                                ;*   :  ENDIF
                                ;*   :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
                                ;*   :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
                                ;*   :  IF 18 BIT DATA SAME AS EXPECTED
                                ;*   :  :  THEN-PROCEED
                                ;*   :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
                                ;*   :  ENDIF
                                ;*   :  POINT TO NEXT SET OF EXPECTED DATA
                                ;*   :  IF ALL DONE WITH CURRENT XFR
                                ;*   :  :  THEN-UPDATE TO NEXT PACKING MODE
                                ;*   :  :     IF ALL MODES DONE
                                ;*   :  :     :  THEN-EXIT
                                ;*   :  :     :  ELSE-CONTINUE
                                ;*   :  :     ENDIF
                                ;*   :  :  ELSE-CONTINUE
                                ;*   :  ENDIF
                                ;*   :  DO UNTIL ALL 'SCLK' CHECKED FOR SELECTED XFR
                                ;*  ENDDO
                                ;*ENDTST
4586                            SE
  (1)                           ; ***************************************************************************
  (1)                           ;*ERRORS
  (1)                           ;*------
                                ;*WMC6 MICRO TEST 02
                                ;*WMC6 MICRO ERROR 07
                                ;*WMC BYTE ASSY FOR READ REVERSE XFR
                                ;*M8959 (WMC) & ALL OTHER BOARDS
                                ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
                                ;*AFTER ISSUEING A READ REVERSE COMMAND TO THE TM78
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*WMC6 MICRO TEST 02
                                ;*WMC6 MICRO ERROR 10
                                ;*WMC BYTE ASSY FOR READ REVERSE XFR
                                ;*M8959 (WMC) & ALL OTHER BOARDS
                                ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER IT SET
                                ;*DURING A READ REVERSE XFR
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*WMC6 MICRO TEST 02
                                ;*WMC6 MICRO ERROR 11
                                ;*WMC BYTE ASSY FOR READ REVERSE XFR
                                ;*M8959 (WMC) & ALL OTHER BOARDS
                                ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*WMC6 MICRO TEST 02
                                        ;*WMC6 MICRO ERROR 12
                                        ;*WMC BYTE ASSY FOR READ REVERSE XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC6 MICRO TEST 02
                                        ;*WMC6 MICRO ERROR 13
                                        ;*WMC BYTE ASSY FOR READ REVERSE XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
1881  4586                              S
 (1)                                    ; ***************************************************************************************
1882
1883  4586                              TEST2:  TESTX   @2
 (1)  4586  3E  02            7.0                     MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4588  CD  03  28       18.0                     CALL    TSET            ;SETUP THE TEST
                                        ;%WMC BYTE ASSY FOR READ REVERSE XFR
                                        ;&M8959 (WMC) & ALL OTHER BOARDS
                                        ;SETUP THE NECESSARY FLAGS FOR READ REVERSE
1888  458B  3A  2E  4A        13.0  T2LP:   LDA     SUNIT           ;GET THE SELECTED UNIT
1889  458E  47                   4.0          MOV     B,A
1890  458F  DB  E0              10.0          IN      INTSTA          ;GET THE PORT SELECT STATUS
1891  4591  E6  80               7.0          ANI     BIT7
1892  4593  B0                   4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1893  4594  D3  E0              10.0          OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
1894  4596  3E  60               7.0          MVI     A,P.LWR!P.LCS   ;GET THE LOOP-COMMAND-STATUS BIT
1895  4598  D3  48              10.0          OUT     PDIAG           ;SELECT IT
1896  459A  AF                   4.0          XRA     A
1897  459B  D3  44              10.0          OUT     TAMT            ;CLEAR THE AMTIE LINES
1898  459D  D3  D2              10.0          OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS
1899  459F  D3  D2              10.0          OUT     TRKENA
1900  45A1  3E  05               7.0          MVI     A,5
1901  45A3  32  2C  4A          13.0          STA     FMTNUM          ;INIT THE PACKING MODE TO IMAGE MODE
1902  45A6  3E  04               7.0          MVI     A,P.RPEN        ;SET READ PATH ENABLE
1903  45A8  D3  4C              10.0          OUT     PENAB           ;NOW
1904  45AA  3E  10               7.0          MVI     A,RDCLK         ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1905  45AC  D3  F0              10.0          OUT     CLKCTL
1906  45AE  3E  FF               7.0          MVI     A,$FF
1907  45B0  32  2F  4A          13.0          STA     REVERSE         ;SET TO DO A READ REVERSE XFR
1908  45B3  AF                   4.0          XRA     A
1909  45B4  D3  C0              10.0          OUT     DBUSCTL         ;INIT THE DATA BUS CONTROL
1910
1911                                    ;INIT THE COUNTER CONTROL AND COUNTERS
1912
1913  45B6                              T2LOOP: INICNT
 (1)  45B6  3E  30            7.0                     MVI     A,@060          ;INIT THE
 (1)  45B8  D3  D7           10.0                     OUT     CNTCTL          ;BYTE COUNTER
```

```
 (1)  45BA  3E  70        7.0                    MVI   A,@160          ;INIT THE
 (1)  45BC  D3  D7       10.0                    OUT   CNTCTL          ;PAD COUNTER
 (1)  45BE  3E  B0        7.0                    MVI   A,@260          ;INIT THE
 (1)  45C0  D3  D7       10.0                    OUT   CNTCTL          ;ERROR COUNTER
1914  45C2                             CLRECT
 (1)  45C2  AF            4.0                    XRA   A               ;CLEAR THE ACCUMULATOR
 (1)  45C3  D3  D6       10.0                    OUT   ERRCNT          ;CLEAR BITS 7-0
 (1)  45C5  D3  D6       10.0                    OUT   ERRCNT          ;CLEAR BITS 15-8
1915  45C7  3A  2C  4A   13.0                    LDA   FMTNUM          ;GET THE SELECTED PACKING MODE #
1916  45CA  87            4.0                    ADD   A               ;DOUBLE THE COUNT - 2 ADDR PER ENTRY IN TABLE
1917  45CB  5F            4.0                    MOV   E,A
1918  45CC  16  00        7.0                    MVI   D,0
1919  45CE  21  BD  48   10.0           T2RR:    LXI   H,EDREV         ;POINT TO READ REVERSE
1920  45D1  19           10.0                    DAD   D               ;GET OFFSET TO EXPECTED DATA
1921  45D2  7E            7.0                    MOV   A,M             ;GET LOW BYTE OF ADDRESS
1922  45D3  32  D2  48   13.0                    STA   EXPADR          ;SAVE IT
1923  45D6  23            6.0                    INX   H               ;POINT TO THE HIGH BYTE
1924  45D7  7E            7.0                    MOV   A,M             ;GET IT
1925  45D8  32  D3  48   13.0                    STA   EXPADR+1        ;SAVE HIGH BYTE OF EXPECTED DATA ADDRESS
1926  45DB  3A  2C  4A   13.0                    LDA   FMTNUM          ;GET THE PACKING MODE # AGAIN
1927  45DE  5F            4.0                    MOV   E,A
1928  45DF  16  00        7.0                    MVI   D,0
1929  45E1  21  CB  48   10.0                    LXI   H,ESCLKT        ;POINT TO TABLE OF EXPECTED SCLKS/MODE
1930  45E4  19           10.0                    DAD   D               ;POINT TO TABLE OFFSET #
1931  45E5  7E            7.0                    MOV   A,M             ;GET THE EXPECTED # SCLKS/MODE SELECTED
1932  45E6  32  2B  4A   13.0                    STA   ESCLK           ;SAVE THE NUMBER FOR THIS PASS
1933  45E9  3A  2C  4A   13.0                    LDA   FMTNUM          ;GET THE PACKING MODE # AGAIN
1934  45EC  07            4.0                    RLC
1935  45ED  07            4.0                    RLC
1936  45EE  07            4.0                    RLC
1937  45EF  07            4.0                    RLC
1938  45F0  E6  70        7.0                    ANI   $70            ;POSITION THE FMT #
1939  45F2  D3  D0       10.0                    OUT   DATACTL         ;LOAD THE DATA CONTROL WORD
1940  45F4                                       ROUT  R02H            ;SAVE FOR ERROR PRINTING
 (1)  45F4  D3  85       10.0                          OUT   R02H              ;WRITE AC INTO R02H
 (1)  45F6  7F            4.0                          MOV   A,A               ;RETRY LINK
1941  45F7  3E  9B        7.0                    MVI   A,@233          ;SET THE DDR CONTROL TO 'IN'
1942  45F9  D3  DB       10.0                    OUT   DDRCTL          ;TO READ THE 'DDR'
1943  45FB  3E  00        7.0                    MVI   A,@0            ;GET THE RESIDUAL CHAR WORD
1944  45FD  D3  D1       10.0                    OUT   RESCHR          ;LOAD IT
1945  45FF  3E  32        7.0                    MVI   A,50            ;SETUP THE BYTE COUNTER FOR 8 DATA GROUPS
1946  4601  D3  D4       10.0                    OUT   BYTCNT
1947  4603  AF            4.0                    XRA   A
1948  4604  D3  D4       10.0                    OUT   BYTCNT
1949  4606  3E  06        7.0                    MVI   A,6               ;NEED 6 PAD CHARACTERS FOR THE XFR
1950  4608  D3  D5       10.0                    OUT   PADCNT          ;LOAD IT
1951  460A  AF            4.0                    XRA   A
1952  460B  D3  D5       10.0                    OUT   PADCNT
1953  460D                                       ROUT  R02L            ;CLEAR OUT THE REST OF REG 2 IN CAS
 (1)  460D  D3  84       10.0                          OUT   R02L              ;WRITE AC INTO R02L
 (1)  460F  7F            4.0                          MOV   A,A               ;RETRY LINK
1954  4610                                       ROUT  R05L            ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
 (1)  4610  D3  8A       10.0                          OUT   R05L              ;WRITE AC INTO R05L
 (1)  4612  7F            4.0                          MOV   A,A               ;RETRY LINK
```

```
1955  4613  32  2A  4A   13.0          STA     SCLKCT          ;INIT THE SCLK DETECTED COUNTER
1956
1957                                ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1958
1959  4616  3E  01       7.0          MVI     A,W.RST         ;GET THE WMC RESTART BIT
1960  4618  D3  D3      10.0          OUT     WMCCTL          ;SET RESTART
1961  461A  3E  15       7.0          MVI     A,SSTEP.RDCLK   ;GET SINGLE STEP FOR WMC
1962  461C  D3  F0      10.0          OUT     CLKCTL          ;SET SINGLE STEP MODE
1963  461E  32  30  4A   13.0          STA     CCTLWD          ;SAVE THE WORD
1964  4621  CD  81  4A   18.0          CALL    CLKSYS          ;SET THE RESTART
1965  4624  CD  81  4A   18.0          CALL    CLKSYS
1966  4627  AF          4.0          XRA     A
1967  4628  D3  D3      10.0          OUT     WMCCTL          ;FINISH THE RESTART CYCLE
1968
1969                                ;CLOCK THE WMC THRU ITS RESTART AREA
1970
1971  462A  06  B4       7.0          MVI     B,180           ;GET A LOOP COUNT
1972  462C  CD  81  4A   18.0  CLKRS2: CALL    CLKSYS          ;CLOCK THE SYSTEM
1973  462F  05           4.0          DCR     B               ;DECREMENT THE LOOP COUNTER
1974  4630  C2  2C  46   10.0          JNZ     CLKRS2          ;CLOCK TILL COUNT RUN OUT
1975
1976  4633  3E  94       7.0          MVI     A,W.GCR!W.ENAB!W.REV    ;SETUP FOR A GCR DATA XFR
1977  4635  D3  D3      10.0  T2ST:   OUT     WMCCTL          ;LOAD THE WORD IN WMC
1978                                ;CALCULATE THE CRC DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1979
1980  4637  21  7F  46   10.0  C2EC1:  LXI     H,CRCG          ;POINT THE THE 1ST DATA GROUP DATA TABLE
1981  463A  06  07       7.0          MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1982  463C  CD  42  4A   18.0          CALL    CLECC           ;INIT THE ECC CHARACTER
1983
1984  463F  7E           7.0  C2EC1L: MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
1985  4640  CD  47  4A   18.0          CALL    ECC             ;CALC ECC WITH THIS CHARACTER
1986  4643  23           6.0          INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1987  4644  05           4.0          DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
1988  4645  C2  3F  46   10.0          JNZ     C2EC1L          ;STAY HERE TILL ALL DATA BYTES USED
1989  4648  3A  6E  4A   13.0          LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1990  464B  32  86  46   13.0          STA     CRCGEC          ;SAVE THE CALCULATED ECC CHARACTER
1991
1992                                ;CALCULATE THE 1ST DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1993
1994  464E  21  9F  48   10.0  C2EC2:  LXI     H,RDG1          ;POINT THE THE 1ST DATA GROUP DATA TABLE
1995  4651  06  07       7.0          MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1996  4653  CD  42  4A   18.0          CALL    CLECC           ;INIT THE ECC CHARACTER
1997
1998  4656  7E           7.0  C2EC2L: MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
1999  4657  CD  47  4A   18.0          CALL    ECC             ;CALC ECC WITH THIS CHARACTER
2000  465A  23           6.0          INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
2001  465B  05           4.0          DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
2002  465C  C2  56  46   10.0          JNZ     C2EC2L          ;STAY HERE TILL ALL DATA BYTES USED
2003  465F  3A  6E  4A   13.0          LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
2004  4662  32  A6  48   13.0          STA     RDG1EC          ;SAVE THE CALCULATED ECC CHARACTER
2005                                ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
2006
2007  4665  21  A7  48   10.0  C2EC3:  LXI     H,RDG2          ;POINT THE THE 2ND DATA GROUP DATA TABLE
2008  4668  06  07       7.0          MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
```

```
2009  466A  CD  42  4A   18.0          CALL   CLECC              ;INIT THE ECC CHARACTER
2010
2011  466D  7E              7.0  C2EC3L: MOV    A,M               ;GET A DATA BYTE FROM THE TABLE
2012  466E  CD  47  4A      18.0          CALL   ECC               ;CALC ECC WITH THIS CHARACTER
2013  4671  23              6.0          INX    H                 ;POINT TO THE NEXT DATA BYTE IN TABLE
2014  4672  05              4.0          DCR    B                 ;DECREMENT THE COUNTER OF DATA BYTES
2015  4673  C2  6D  46     10.0          JNZ    C2EC3L            ;STAY HERE TILL ALL DATA BYTES USED
2016  4676  3A  6E  4A     13.0          LDA    ECCCHR            ;GET THE CALCULATED ECC CHARACTER
2017  4679  32  AE  48     13.0          STA    RDG2EC            ;SAVE THE CALCULATED ECC CHARACTER
2018  467C  C3  87  46     10.0          JMP    T2PLO
2019
2020  467F                       RESG:
2021  467F  00                   CRCG:  .BYTE  0                 ;CRC GROUP
2022  4680  00                          .BYTE  0                 ;CRC GROUP 'CRC' CHARACTER
2023  4681  00                          .BYTE  0                 ;CRC GROUP 'CRC' CHAR
2024  4682  00                          .BYTE  0                 ;CRC GROUP 'CRC' CHAR
2025
2026  4683                       RESG2:
2027  4683  00                   CRCG2: .BYTE  0
2028  4684  00                          .BYTE  0
2029  4685  00                          .BYTE  0                 ;RESIDUAL CHARACTER
2030  4686  00                   CRCGEC: .BYTE 0
2031  4687  3E  A8              7.0  T2PLO: MVI    A,R.PLO1!R.TBJN!R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2032                                                                ;SET TIE BUS JAM AND PLO DISABLE
2033  4689  D3  09             10.0          OUT    RPCTL
2034  468B  AF                4.0          XRA    A
2035  468C  D3  0A             10.0          OUT    RTIEB             ;CLEAR THE TIE BUS
2036  468E  D3  08             10.0          OUT    RFIFOL            ;CLOCK THE FIFO'S
2037  4690  3E  0D              7.0          MVI    A,RCLRT           ;ISSUE CLEAR ALL COMMAND
2038  4692  D3  0B             10.0          OUT    RCMD
2039
2040  4694  00                4.0          NOP                      ;WAIT
2041  4695  00                4.0          NOP                      ;WAIT
2042  4696  00                4.0          NOP
2043  4697  00                4.0          NOP
2044  4698  00                4.0          NOP
2045
2046  4699  3E  A9              7.0          MVI    A,R.PLO1!R.STPC.R.PLOD!R.TBJN   ;STOP THE READ PATH
2047                                                                ;SET TIE BUS JAM AND PLO DISABLE
2048  469B  D3  09             10.0          OUT    RPCTL
2049
2050  469D  21  7F  46         10.0  T2SG1:  LXI    H,CRCG            ;POINT TO THE CRC DATA GROUP
2051  46A0  CD  43  48         18.0          CALL   CONVRT            ;CONVERT IT TO 4X5
2052  46A3  CD  58  48         18.0          CALL   LOADIT            ;PUT IT IN THE FIFO
2053
2054  46A6  21  83  46         10.0  T2SG2:  LXI    H,CRCG2           ;POINT TO 2ND HALF OF CRC GROUP
2055  46A9  CD  43  48         18.0          CALL   CONVRT            ;CHANGE IT
2056  46AC  CD  58  48         18.0          CALL   LOADIT            ;AND PUT IT IN THE FIFO
2057
2058  46AF  21  7F  46         10.0  T2SG3:  LXI    H,RESG            ;POINT TO THE RESIDUAL DATA GROUP
2059  46B2  CD  43  48         18.0          CALL   CONVRT            ;CHANGE IT
2060  46B5  CD  58  48         18.0          CALL   LOADIT            ;AND PUT IT IN THE FIFO
2061
2062  46B8  21  83  46         10.0  T2SG4:  LXI    H,RESG2           ;POINT TO THE 2ND HALF OF RESIDUAL DATA
```

```
2063  46BB  CD  43  48      18.0            CALL    CONVRT          ;CHANGE IT
2064  46BE  CD  58  48      18.0            CALL    LOADIT          ;AND PUT IT IN THE FIFO
2065
2066  46C1  21  35  4A       10.0  T2EMK:    LXI     H,TRNOUT        ;SETUP TO LOAD THE END MARK SIGNAL
2067  46C4  06  05            7.0            MVI     B,5
2068  46C6  3E  FF            7.0  T2EMKL:   MVI     A,$FF
2069  46C8  77                7.0            MOV     M,A             ;SAVE A CHAR
2070  46C9  23                6.0            INX     H               ;POINT TO PARITY BIT STORAGE
2071  46CA  3E  01            7.0            MVI     A,1
2072  46CC  77                7.0            MOV     M,A
2073  46CD  23                6.0            INX     H
2074  46CE  05                4.0            DCR     B               ;DECREMENT LOOP COUNTER
2075  46CF  C2  C6  46       10.0            JNZ     T2EMKL          ;LOOP TILL END MARK IN 4X5 STORAGE AREA
2076  46D2  CD  58  48       18.0            CALL    LOADIT          ;LOAD THE END MARK INTO FIFO
2077  46D5  21  A3  48       10.0  T2SGD2:   LXI     H,RSG2          ;POINT TO THE 2ND SUBGROUP OF DATA
2078  46D8  CD  43  48       18.0            CALL    CONVRT          ;MAKE IT 4X5
2079  46DB  CD  58  48       18.0            CALL    LOADIT          ;INJECT IT INTO FIFO
2080
2081  46DE  21  9F  48       10.0  T2SGD1:   LXI     H,RDG1          ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
2082  46E1  CD  43  48       18.0            CALL    CONVRT          ;CONVERT SUBGROUP - 4X5
2083  46E4  CD  58  48       18.0            CALL    LOADIT          ;INJECT THE DATA INTO THE FIFO
2084
2085  46E7  21  AB  48       10.0  T2SGD4:   LXI     H,RSG4          ;POINT TO THE 4TH SUBGROUP OF DATA
2086  46EA  CD  43  48       18.0            CALL    CONVRT          ;MAKE IT 4X5
2087  46FD  CD  58  48       18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2088
2089  46F0  21  A7  48       10.0  T2SGD3:   LXI     H,RDG2          ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
2090  46F3  CD  43  48       18.0            CALL    CONVRT          ;MAKE IT 4X5
2091  46F6  CD  58  48       18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2092  46F9  CD  58  48       18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2093  46FC  CD  58  48       18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2094  46FF  AF                4.0            XRA     A               ;CLEAR THE LOAD MORE FLAG
2095  4700  32  41  4A       13.0            STA     MOREFG          ;SO PGM WILL LOAD ANOTHER GROUP ON 1ST
2096                                                                  ;SCLK
2097
2098  4703  3E  0B            7.0            MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
2099  4705  D3  0B           10.0            OUT     RCMD            ;COMMAND
2100
2101  4707  11  01  00       10.0  T2CLKA:   LXI     D,1             ;SET WATCH DOG INCREMENT
2102  470A  21  68  C5       10.0            LXI     H,-15000        ;SET WATCH DOG COUNT TO 15000
2103  470D  D3  0C           10.0  T2CLK:    OUT     RINST           ;STEP THE READ PATH
2104  470F  CD  81  4A       18.0            CALL    CLKSYS          ;CLOCK THE WMC
2105  4712  DB  C0           10.0            IN      DBUSSTA         ;GET THE DATABUS STATUS
2106  4714  E6  02            7.0            ANI     T.SCLK          ;SEE IF SCLK UP - DATA READY
2107  4716  C2  2B  47       10.0            JNZ     T2SCLK          ;JUMP IF SCLK UP
2108  4719  19               10.0            DAD     D
2109  471A  D2  0D  47       10.0            JNC     T2CLK           ;STAY IN LOOP UNLESS TIMFOUT
2110
```

```
2112  471D                                      ERR     T2LOOP,DUMM3,3
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  471D  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4720  07                                           .BYTE   MSGN            ;MESSAGE NUMBER :D
 (1)  4721  03                                           .BYTE   3
 (1)  4722  CD  15  28        18.0               DUMM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4725  DA  B6  45        10.0                       JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2113                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
2114                                    ;>AFTER ISSUEING A READ REVERSE COMMAND TO THE TM78
2115                                    ;<FATAL ERROR - TEST ABORTED
2116  4728  C3  6F  45        10.0              JMP     EXIT
2117
2118                                    ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO -0
2119                                    ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
2120                                    ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDICATING THE DATA IS
2121                                    ;ON THE MASSBUS
2122
2123  472B  11  01  00        10.0  T2SCLK: LXI     D,1             ;SETUP THE DROP DEAD LOOP COUNTER
2124  472E  21  68  C5        10.0          LXI     H,-15000        ;SO IT WON'T HANG FOREVER
2125  4731  3A  41  4A        13.0          LDA     MOREFG          ;SEE IF NEED MORE DATA
2126  4734  A7                 4.0          ANA     A               ;WILL ONLY LOAD ANOTHER GROUP IF 1ST SCLK
2127  4735  C2  43  47        10.0          JNZ     T2SCKL          ;JUMP OVER IF NOT 1ST TIME
2128  4738  CD  58  48        18.0          CALL    LOADIT
2129  473B  CD  58  48        18.0          CALL    LOADIT          ;PUT IN ANOTHER DATA GROUP - DUMMY DATA
2130  473E  3E  FF             7.0          MVI     A,$FF
2131  4740  32  41  4A        13.0          STA     MOREFG          ;SET THE DATA FLAG
2132  4743  D3  0C             10.0  T2SCKL: OUT     RINST           ;CLOCK THE READ PATH ONCE
2133  4745  CD  81  4A        18.0          CALL    CLKSYS          ;CLOCK THE WMC TO CLEAR 'SCLK'
2134  4748  DB  C0             10.0          IN      DBUSSTA         ;SEE IF 'SCLK' IS GONE
2135  474A  E6  02              7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
2136  474C  CA  61  47        10.0          JZ      T2SCK0          ;JUMP IF EVER GOES =0
2137  474F  19                 10.0          DAD     D               ;DECREMENT THE LOOP COUNTER
2138  4750  D2  43  47        10.0          JNC     T2SCKL          ;& LOOP TILL DEAD
2139  4753                                  ERR     T2LOOP,DUMM4,3
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4753  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                         MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4756  08                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4757  03                                           .BYTE   3
 (1)  4758  CD  15  28        18.0               DUMM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  475B  DA  B6  45        10.0                       JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2140                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER IT SET
2141                                    ;>DURING A READ REVERSE XFR
2142                                    ;<FATAL ERROR - TEST ABORTED
2143  475E  C3  29  48        10.0              JMP     EXIT2
2144                                    ;HERE WHEN 'SCLK' CLEARED - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
2145                                    ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
2146
2147  4761  DB  D8             10.0  T2SCK0: IN      DDRA            ;GET THE DDR A DATA
2148  4763  32  27  4A        13.0          STA     SDDRA           ;SAVE IT
2149  4766                                  ROUT    R16L            ;SAVE FOR PRINTING
 (1)  4766  D3  9C             10.0                       OUT     R16L                    ;WRITE AC INTO R16L
 (1)  4768  7F                  4.0                       MOV     A,A                     ;RETRY LINK
```

```
2150  4769  DB  D9        10.0          IN    DDRB          ;GET DDR B DATA
2151  476B  32  28  4A    13.0          STA   SDDRB         ;SAVE IT
2152  476E                              ROUT  R16H          ;SAVE FOR ERROR REPORTING
 (1)  476E  D3  9D        10.0                OUT   R16H            ;WRITE AC INTO R16H
 (1)  4770  7F             4.0                MOV   A,A             ;RETRY LINK
2153  4771  DB  DA        10.0          IN    DDRC          ;GET DDR C DATA
2154  4773  E6  03         7.0          ANI   3
2155  4775  32  29  4A    13.0          STA   SDDRC         ;SAVE IT
2156  4778  F6  80         7.0          ORI   BIT7          ;ADD IN THE PRINT FLAG
2157  477A                              ROUT  R17L          ;SAVE IT FOR PRINTING
 (1)  477A  D3  9E        10.0                OUT   R17L            ;WRITE AC INTO R17L
 (1)  477C  7F             4.0                MOV   A,A             ;RETRY LINK
2158
2159  477D  3A  2A  4A    13.0          LDA   SCLKCT        ;GET THE CURRENT SCLK COUNT
2160  4780  3C             4.0          INR   A             ;COUNT THIS SCLK
2161  4781  32  2A  4A    13.0          STA   SCLKCT        ;SAVE THE NEW COUNT
2162  4784                              ROUT  R05L          ;ALSO IN CAS FOR ERROR REPORTING
 (1)  4784  D3  8A        10.0                OUT   R05L            ;WRITE AC INTO R05L
 (1)  4786  7F             4.0                MOV   A,A             ;RETRY LINK
2163  4787  AF             4.0          XRA   A
2164  4788                              ROUT  R05H
 (1)  4788  D3  8B        10.0                OUT   R05H            ;WRITE AC INTO R05H
 (1)  478A  7F             4.0                MOV   A,A             ;RETRY LINK
2165
2166                                  ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED
2167
2168  478B  2A  D2  48    16.0  T2CKA:  LHLD  EXPADR          ;POINT TO THE TABLE OF EXPECTED DATA
2169  478E  7E             7.0          MOV   A,M           ;GET BITS 7-0
2170  478F                              ROUT  R14L          ;SAVE FOR ERROR REPORTING
 (1)  478F  D3  98        10.0                OUT   R14L            ;WRITE AC INTO R14L
 (1)  4791  7F             4.0                MOV   A,A             ;RETRY LINK
2171  4792  23             6.0          INX   H             ;POINT TO NEXT BYTE
2172  4793  7E             7.0          MOV   A,M           ;GET BITS 15-8
2173  4794                              ROUT  R14H          ;SAVE IT
 (1)  4794  D3  99        10.0                OUT   R14H            ;WRITE AC INTO R14H
 (1)  4796  7F             4.0                MOV   A,A             ;RETRY LINK
2174  4797  23             6.0          INX   H
2175  4798  7E             7.0          MOV   A,M           ;GET BITS 17-16
2176  4799  F6  80         7.0          ORI   BIT7          ;ADD IN THE PRINT FLAG
2177  479B                              ROUT  R15L          ;SAVE FOR ERROR REPORT
 (1)  479B  D3  9A        10.0                OUT   R15L            ;WRITE AC INTO R15L
 (1)  479D  7F             4.0                MOV   A,A             ;RETRY LINK
2178  479E  3A  27  4A    13.0          LDA   SDDRA         ;GET THE DDR A DATA
2179  47A1                              ROUT  ADATA         ;SAVE IN CASE OF ERROR DETECTED
 (1)  47A1  D3  94        10.0                OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  47A3  7F             4.0                MOV   A,A             ;RETRY LINK
2180  47A4  2A  D2  48    16.0          LHLD  EXPADR        ;POINT TO THE EXPECTED DATA TABLE
2181  47A7  BE             7.0          CMP   M             ;ACTUAL DATA = EXPECTED?
2182  47A8  CA  B4  47    10.0          JZ    T2CKAC        ;JUMP IF OK
2183  47AB  7E             7.0          MOV   A,M           ;GET THE EXPECTED DATA
```

```
2185  47AC                                          ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  47AC   D3   95          10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47AE   7F                4.0                         MOV     A,A             ;RETRY LINK
2186  47AF                                          ERRB    T2LOOP,T2CKAC,@10
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  47AF   CD   12   28     18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)         0009                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47B2   09                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47B3   0E                                           .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  47B4   CF   15   28     18.0                 T2CKAC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47B7   DA   B6   45     10.0                         JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2187                                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
2188  47BA   23                 6.0           T2CKB:  INX     H               ;POINT TO THE EXPECTED DATA FOR DDR B
2189  47BB   3A   28   4A       13.0                  LDA     SDDRB           ;GET THE DDR B DATA
2190  47BE                                          ROUT    ADATA           ;SAVE FOR ERROR REPORT
 (1)  47BE   D3   94          10.0                         OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47C0   7F                4.0                         MOV     A,A             ;RETRY LINK
2191  47C1   BE                 7.0                  CMP     M               ;ACTUAL = EXPECTED DATA?
2192  47C2   CA   CE   47       10.0                  JZ      T2CKBC          ;JUMP IF OK
2193  47C5   7E                 7.0                  MOV     A,M             ;GET THE EXPECTED DATA
2194  47C6                                          ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  47C6   D3   95          10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47C8   7F                4.0                         MOV     A,A             ;RETRY LINK
2195  47C9                                          ERRB    T2LOOP,T2CKBC,@10
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  47C9   CD   12   28     18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)         000A                                         MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47CC   0A                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47CD   08                                           .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  47CE   CD   15   28     18.0                 T2CKBC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47D1   DA   B6   45     10.0                         JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2196                                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
2197
2198  47D4   23                 6.0           T2CKC:  INX     H               ;POINT TO EXPECTED DDR C DATA
2199  47D5   3A   29   4A       13.0                  LDA     SDDRC           ;GET DDR C DATA
2200  47D8                                          ROUT    ADATA           ;SAVE FOR ERROR REPORT
 (1)  47D8   D3   94          10.0                         OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47DA   7F                4.0                         MOV     A,A             ;RETRY LINK
2201  47DB   BE                 7.0                  CMP     M               ;ACTUAL DATA = EXPECTED DDRC DATA?
2202  47DC   CA   E8   47       10.0                  JZ      T2CKCC          ;JUMP IF OK
2203  47DF   7E                 7.0                  MOV     A,M             ;GET THE EXPECTED DATA
2204  47E0                                          ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  47E0   D3   95          10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47E2   7F                4.0                         MOV     A,A             ;RETRY LINK
2205  47E3                                          ERRB    T2LOOP,T2CKCC,@10
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  47E3   CD   12   28     18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)         0008                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47E6   08                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47E7   08                                           .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  47E8   CD   15   28     18.0                 T2CKCC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47EB   DA   B6   45     10.0                         JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2206                                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
2207
```

```
2208  47EE  2A  D2  48   16.0          LHLD   EXPADR          ;GET THE CURRENT EXPECTED DATA ADDR.
2209  47F1  23               6.0       INX    H
2210  47F2  23               6.0       INX    H               ;UPDATE THE EXPECTED ADDRESS POINTER
2211  47F3  23               6.0       INX    H               ;TO THE DATA
2212  47F4  22  D2  48   16.0          SHLD   EXPADR          ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
2213  47F7  3A  2A  4A   13.0          LDA    SCLKCT          ;GET THE SCLK COUNT
2214  47FA  21  2B  4A   10.0          LXI    H,ESCLK         ;POINT TO THE EXPECTED SCLK COUNT
2215  47FD  BE               7.0       CMP    M               ;DONE WITH THIS TEST PASS?
2216  47FE  C2  07  47    0.0          JNZ    T2CLKA          ;JUMP IF STILL EXPECTING MORE SCLKS
2217
2218                                ;UPDATE THE PACKING MODE NUMBER FOR THE NEXT PASS
2219
2220  4801  3A  2C  4A   13.0          LDA    FMTNUM          ;GET THE CURRENT NUMBER
2221  4804  3C               4.0       INR    A
2222  4805  32  2C  4A   13.0          STA    FMTNUM          ;SAVE THE NEW NUMBER
2223  4808  FE  05               7.0       CPI    5               ;READY TO DO MODE #6??
2224  480A  C2  15  48   10.0          JNZ    T2PM6           ;SEE IF TIME TO DO MODE #0
2225  480D  3E  06               7.0       MVI    A,6             ;YUP (ALREADY DID #5 ON 1ST PASS)
2226  480F  32  2C  4A   13.0          STA    FMTNUM          ;SAVE IT
2227  4812  C3  B6  45   10.0          JMP    T2LOOP          ;DO MODE 6
2228  4815  FE  06               7.0  T2PM6: CPI    6               ;READY TO START AT MODE 0
2229  4817  CA  22  48   10.0          JZ     T2PM0           ;YUP
2230  481A  FE  07               7.0       CPI    7               ;DONE WITH ALL THE PACKING MODES?
2231  481C  CA  29  48   10.0          JZ     EXIT2           ;YUP
2232  481F  C3  B6  45   10.0          JMP    T2LOOP          ;DO THE NEXT PACKING MODE
2233
2234  4822  AF               4.0  T2PM0: XRA    A               ;START AT MODE 0
2235  4823  32  2C  4A   13.0          STA    FMTNUM          ;SAVE THE PACKING MODE #
2236  4826  C3  B6  45   10.0          JMP    T2LOOP          ;DO THE MODE
2237
2238  4829                        EXIT2:  ENDTST  T2LP
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4829                               REQ    7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4829  CD  06  28   18.0                            CALL   REQST
 (2)  482C  00                           .BYTE                  ;DATA PATTERN NUMBER
 (2)  482D  00  00                       .WORD                  ;SYSTEM '*' COUNT
 (2)  482F  00  00                       .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4831  00                           .BYTE                  ;DATA COMPARE FLAG IF =1
 (2)  4832  07                           .BYTE  7               ;REQUEST CODE
 (1)  4833  3A  9A  4F   13.0          LDA    ITERA           ;GET ITERATION COUNT
 (1)  4836  3D               4.0       DCR    A               ;DOWNCOUNT
 (1)  4837  32  9A  4F   13.0          STA    ITERA           ;SAVE COUNT
 (1)  483A  F2  8B  45   10.0          JP     T2LP            ;DO TEST UNTIL TILL = 0
2239  483D  CD  7C  49   18.0          CALL   CLEAR           ;CLEAR ALL TU PORTS
2240  4840  C3  18  28   10.0          JMP    TSTEND          ;ALL DONE!
2241                                ;HERE IS THE SUBROUTINES NECESSARY FOR CONVERTING SUBGROUPS AND LOADING
2242                                ;THEM INTO THE FIFO
2243
2244  4843                          S
 (1)                                ;*********************************************************************************
2245                                ;*CONVRT -- THIS SUBROUTINE WILL GET A SUBGROUP OF DATA AS POINTED TO BY
2246                                ;*      THE H & L REGISTERS AND CAUSE THEM TO BE TRANSLATED (4X5). THE
2247                                ;*      TRANSLATED DATA WILL BE IN A TABLE CALLED 'TRNOUT'.
2248  4843                          S
```

```
 (1)                                    ; ***************************************************************************
2249
2250  4843  D5                  12.0  CONVRT: PUSH    D               ;SAVE D & E
2251  4844  11  31  4A           10.0          LXI     D,TRNIN         ;POINT TO THE TEMP STORAGE FOR DATA BYTES
2252  4847  06  04                7.0          MVI     B,4             ;SETUP A LOOP COUNTER FOR 4 BYTES
2253
2254  4849  7E                    7.0  CNV1:   MOV     A,M             ;GET A BYTE OF DATA
2255  484A  EB                    4.0          XCHG                    ;POINT TO THE STORAGE
2256  484B  77                    7.0          MOV     M,A             ;SAVE DATA BYTE IN TEMP STORAGE
2257  484C  EB                    4.0          XCHG                    ;RESET POINTER
2258  484D  23                    6.0          INX     H               ;POINT TO NEXT DATA BYTE
2259  484E  13                    6.0          INX     D               ;POINT TO NEXT TEMP STORAGE ADDRESS
2260  484F  05                    4.0          DCR     B               ;UPDATE THE LOOP COUNT
2261  4850  C2  49  48           10.0          JNZ     CNV1            ;LOOP TILL 4 BYTES IN TEMP STORAGE
2262  4853  CD  A7  49           18.0          CALL    T4X5            ;CONVERT THE 'SUBGROUP' OF DATA
2263  4856  D1                  10.0          POP     D
2264  4857  C9                  10.0          RET
2265  4858                              S
 (1)                                    ; ***************************************************************************
2266                                    ;*LOADIT -- THIS SUBROUTINE WILL TAKE THE TRANSLATED SUBGROUP (4X5)
2267                                    ;*      DATA FROM TABLE 'TRNOUT' AND LOAD THE DATA INTO THE READ
2268                                    ;*      PATH FIFO.
2269  4858                              S
 (1)                                    ; ***************************************************************************
2270
2271  4858  3A  2F  4A           13.0  LOADIT: LDA     REVERSE         ;SEE IF A READ REVERSE XFR
2272  485B  A7                    4.0          ANA     A
2273  485C  C2  76  48           10.0          JNZ     RLOAD           ;REVERSE LOAD INTO FIFO
2274  485F  21  35  4A           10.0  FLOAD:  LXI     H,TRNOUT        ;POINT TO THE TRANSLATED DATA TABLE
2275  4862  06  05                7.0          MVI     B,5             ;GET THE LOOP COUNT FOR SUBGROUP
2276
2277  4864  7E                    7.0  LDFIFO: MOV     A,M             ;GET A BYTE OF DATA
2278  4865  D3  40                10.0          OUT     TCMD            ;STORE IT IN THE CMD REG FOR LOADING
2279  4867  23                    6.0          INX     H               ;POINT TO THE PARITY BIT
2280  4868  7E                    7.0          MOV     A,M             ;GET THE PARITY BIT
2281  4869  07                    4.0          RLC
2282  486A  F6  60                7.0          ORI     P.LWR!P.LCS     ;ADD IN THE OTHER NECESSARY BITS
2283  486C  D3  48                10.0          OUT     PDIAG           ;LOAD INTO CONTROL REG FOR LOADING
2284  486E  D3  08                10.0          OUT     RFIFOL          ;STROBE THE DATA INTO FIFO
2285  4870  23                    6.0          INX     H               ;POINT TO THE NEXT SET OF DATA
2286  4871  05                    4.0          DCR     B               ;DECREMENT THE LOOP COUNTER
2287  4872  C2  64  48           10.0          JNZ     LDFIFO          ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
2288  4875  C9                  10.0          RET
2289
2290  4876  21  3D  4A           10.0  RLOAD:  LXI     H,TRNOUT+8      ;POINT TO LAST ENTRY OF SUBGROUP
2291  4879  06  05                7.0          MVI     B,5             ;GET THE LOOP COUNT NEEDED
2292
2293  487B  7E                    7.0  RLFIFO: MOV     A,M             ;GET A BYTE OF DATA
2294  487C  D3  40                10.0          OUT     TCMD            ;STORE IT IN THE CMD REG FOR LOADING
2295  487E  23                    6.0          INX     H               ;POINT TO THE PARITY BIT
2296  487F  7E                    7.0          MOV     A,M             ;GET THE PARITY BIT
2297  4880  07                    4.0          RLC
2298  4881  F6  60                7.0          ORI     P.LWR!P.LCS     ;ADD IN THE OTHER NECESSARY BITS
2299  4883  D3  48                10.0          OUT     PDIAG           ;LOAD INTO CONTROL REG FOR LOADING
```

```
2300  4885  D3  08          10.0          OUT   RFIFOL          ;STROBE THE DATA INTO FIFO
2301  4887  2B               6.0          DCX   H               ;BACK THE POINTER UP TO THE
2302  4888  2B               6.0          DCX   H               ;BEGINING OF THE PREVIOUS
2303  4889  2B               6.0          DCX   H               ;ENTRY PAIR
2304  488A  05               4.0          DCR   B               ;DECREMENT THE LOOP COUNTER
2305  488B  C2  7B  48       10.0          JNZ   RLFIFO          ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
2306  488E  C9               10.0          RET
2307                                    ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
2308                                    ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ FORWARD'
2309
2310  488F  01                        DG1:    .BYTE   @1      ;1ST BYTE IN THE 1ST SUBGROUP
2311  4890  02                                .BYTE   @2      ;1ST SUBGROUP HAS 4 DATA BYTES IN IT
2312  4891  03                                .BYTE   @3
2313  4892  04                                .BYTE   @4      ;LAST BYTE IN THE 1ST SUBGROUP
2314
2315  4893  05                        SG2:    .BYTE   @5      ;1ST BYTE IN 2ND SUBGROUP
2316  4894  06                                .BYTE   @6      ;2ND SUBGROUP HAS 3 BYTES IN IT TO FINISH THE DATA
2317  4895  07                                .BYTE   @7      ;DATA GROUP OF 7 DATA BYTES
2318  4896  00                        DG1ECC: .BYTE   0       ;STORAGE FOR THE 1ST ECC BYTE (CALCULATED)
2319
2320  4897  08                        DG2:    .BYTE   @10     ;1ST BYTE OF THE 3RD SUBGROUP
2321  4898  09                                .BYTE   @11     ;3RD SUBGROUP HAS 4 DATA BYTES
2322  4899  0A                                .BYTE   @12
2323  489A  0B                                .BYTE   @13     ;LAST DATA BYTE IN 3RD SUBGROUP
2324
2325  489B  0C                        SG4:    .BYTE   @14     ;1ST BYTE IN THE 4TH SUBGROUP
2326  489C  0D                                .BYTE   @15     ;THIS SUBGROUP HAS ONLY 3 DATA BYTES
2327  489D  0E                                .BYTE   @16     ;LAST BYTE IN 4TH SUBGROUP
2328  489E  00                        DG2ECC: .BYTE   0       ;2ND DATA GROUP ECC BYTE STORAGE (CALCULATED)
2329
2330                                    ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
2331                                    ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ REVERSE'
2332
2333  489F  07                        RDG1:   .BYTE   @7      ;1ST BYTE IN THE 1ST SUBGROUP
2334  48A0  06                                .BYTE   @6      ;1ST SUBGROUP HAS 4 DATA BYTES IN IT
2335  48A1  05                                .BYTE   @5
2336  48A2  04                                .BYTE   @4      ;LAST BYTE IN THE 1ST SUBGROUP
2337
2338  48A3  03                        RSG2:   .BYTE   @3      ;1ST BYTE IN 2ND SUBGROUP
2339  48A4  02                                .BYTE   @2      ;2ND SUBGROUP HAS 3 BYTES IN IT TO FINISH THE DATA
2340  48A5  01                                .BYTE   @1      ;DATA GROUP OF 7 DATA BYTES
2341  48A6  00                        RDG1EC: .BYTE   0       ;STORAGE FOR THE 1ST ECC BYTE (CALCULATED)
2342
2343  48A7  0E                        RDG2:   .BYTE   @16     ;1ST BYTE OF THE 3RD SUBGROUP
2344  48A8  0D                                .BYTE   @15     ;3RD SUBGROUP HAS 4 DATA BYTES
2345  48A9  0C                                .BYTE   @14
2346  48AA  0B                                .BYTE   @13     ;LAST DATA BYTE IN 3RD SUBGROUP
2347
2348  48AB  0A                        RSG4:   .BYTE   @12     ;1ST BYTE IN THE 4TH SUBGROUP
2349  48AC  09                                .BYTE   @11     ;THIS SUBGROUP HAS ONLY 3 DATA BYTES
2350  48AD  08                                .BYTE   @10     ;LAST BYTE IN 4TH SUBGROUP
2351  48AE  00                        RDG2EC: .BYTE   0       ;2ND DATA GROUP ECC BYTE STORAGE (CALCULATED)
2352                                    ;THIS IS THE TABLE OF EXPECTED DATA ADDRESSES FOR SELECTED PACKING MODES
2353                                    ;THE DDR REGISTERS WILL BE EXPECTED TO CONTAIN THE DATA IN THESE TABLES
```

```
                                        ;ON EVERY OCCURANCE OF 'SCLK'


                                        ;READ FORWARD EXPECTED DATA TABLES

48AF  D4  48                            EDFWD:  .ADDR   FD11N    ;FWD DATA FOR 11-NORMAL MODE #0
48B1  E9  48                                    .ADDR   FD15N    ;FWD DATA FOR 15-NORMAL MODE #1
48B3  FE  48                                    .ADDR   FD10C    ;FWD DATA FOR 10-COMPAT MODE #2
48B5  22  49                                    .ADDR   FD10D    ;FWD DATA FOR 10-DUMP MODE #3
48B7  3A  49                                    .ADDR   FD10HC   ;FWD DATA FOR 10-HI DEN COMPAT MODE #4
48B9  52  49                                    .ADDR   FDIMAG   ;FWD DATA FOR IMAGE MODE #5
48B8  3A  49                                    .ADDR   FD10HD   ;FWD DATA FOR 10-HI DEN DUMP MODE #6

                                        ;READ REVERSE EXPECTED DATA TABLES

48BD  E9  48                            EDREV:  .ADDR   RD11N    ;REV DATA FOR 11-NORMAL MODE #0
48BF  D4  48                                    .ADDR   RD15N    ;REV DATA FOR 15-NORMAL MODE #1
48C1  10  49                                    .ADDR   RD10C    ;REV DATA FOR 10-COMPAT MODE #2
48C3  2E  49                                    .ADDR   RD10D    ;REV DATA FOR 10-DUMP MODE #3
48C5  46  49                                    .ADDR   RD10HC   ;REV DATA FOR 10-HI DEN COMPAT MODE #4
48C7  52  49                                    .ADDR   RDIMAG   ;REV DATA FOR IMAGE MODE #5
48C9  46  49                                    .ADDR   RD10HD   ;REV DATA FOR 10-HI DEN DUMP MODE #6

                                        ;THIS IS THE TABLE OF EXPECTED SCLK ASSEMBLIES FOR THE SELECTED PACKING
                                        ;MODES

48CB  07                                ESCLKT: .BYTE   7        ;MODE #0 - 11-NORMAL 7 SCLKS
48CC  07                                        .BYTE   7        ;        1 - 15-NORMAL 7 SCLKS
48CD  06                                        .BYTE   6        ;        2 - 10-COMPAT 6 SCLKS (3 36 BIT WORDS)
48CE  04                                        .BYTE   4        ;        3 - 10-DUMP 4 SCLKS (2 36 BIT WORDS)
48CF  04                                        .BYTE   4        ;        4 - 10-HDC 4 SCLKS (2 36 BIT WORDS)
48D0  0E                                        .BYTE   14       ;        5 - IMAGE 14 SCLKS (14 8 BIT WORDS)
48D1  04                                        .BYTE   4        ;        6 - 10-HDD 4 SCLKS (2 36 BIT WORDS)

48D2  00  00                            EXPADR: .ADDR   0        ;ADDRESS OF THE  EXPECTED 'SCLK' DATA TABLE
                                        ;TABLE OF EXPECTED DATA FOR 11-NORMAL BYTE ASSY FOR A READ FWD FUNCTION
                                        ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
                                        ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).

                                        ;THIS IS PACKING MODE #0 - READ FORWARD ... ALSO 15-NORMAL IF READ REV.

48D4                                    RD15N:
48D4  04                                FD11N:  .BYTE   @4       ;EXPECTED DATA -        DDR A (XFR HAS 7 16.)
48D5  08                                        .BYTE   @10      ;1ST SCLK -             DDR B (BIT WORDS OR)
48D6  00                                        .BYTE   @0       ;                       DDR C (14 DATA BYTES)
48D7  0C                                        .BYTE   @14      ;2ND SCLK               DDR A (DATA IS 1 TO 16(8))
48D8  10                                        .BYTE   @20      ;                       DDR B
48D9  00                                        .BYTE   @0       ;                       DDR C
48DA  14                                        .BYTE   @24      ;3RD SCLK
48DB  18                                        .BYTE   @30
48DC  00                                        .BYTE   @0
48DD  1C                                        .BYTE   @34      ;4TH SCLK
48DE  20                                        .BYTE   @40
48DF  00                                        .BYTE   @0
```

```
2408  48E0  24                        .BYTE   @44     ;5TH SCLK
2409  48E1  28                        .BYTE   @50
2410  48E2  00                        .BYTE   @0
2411  48E3  2C                        .BYTE   @54     ;6TH SCLK
2412  48E4  30                        .BYTE   @60
2413  48E5  00                        .BYTE   @0
2414  48E6  34                        .BYTE   @64     ;7TH & LAST SCLK
2415  48E7  38                        .BYTE   @70
2416  48E8  00                        .BYTE   @0
2417                          ;TABLE OF EXPECTED DATA FOR 15-NORMAL BYTE ASSY FOR A READ FWD FUNCTION
2418                          ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2419                          ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2420
2421                          ;THIS IS PACKING MODE #1 - READ FORWARD ... ALSO 11-NORMAL IF READ REV.
2422
2423  48E9                    RD11N:
2424  48E9  08                FD15N:  .BYTE   @10     ;EXPECTED DATA -        DDR A (XFR HAS 7 16.)
2425  48EA  04                        .BYTE   @4      ;1ST SCLK -             DDR B (BIT WORDS OR)
2426  48EB  00                        .BYTE   @0      ;                       DDR C (14 DATA BYTES)
2427  48EC  10                        .BYTE   @20     ;2ND SCLK               DDR A (DATA IS 1 TO 16(8))
2428  48ED  0C                        .BYTE   @14     ;                       DDR B
2429  48EE  00                        .BYTE   @0      ;                       DDR C
2430  48EF  18                        .BYTE   @30     ;3RD SCLK
2431  48F0  14                        .BYTE   @24
2432  48F1  00                        .BYTE   @0
2433  48F2  20                        .BYTE   @40     ;4TH SCLK
2434  48F3  1C                        .BYTE   @34
2435  48F4  00                        .BYTE   @0
2436  48F5  28                        .BYTE   @50     ;5TH SCLK
2437  48F6  24                        .BYTE   @44
2438  48F7  00                        .BYTE   @0
2439  48F8  30                        .BYTE   @60     ;6TH SCLK
2440  48F9  2C                        .BYTE   @54
2441  48FA  00                        .BYTE   @0
2442  48FB  38                        .BYTE   @70     ;7TH & LAST SCLK
2443  48FC  34                        .BYTE   @64
2444  48FD  00                        .BYTE   @0
2445                          ;TABLE OF EXPECTED DATA FOR 10-COMPATABILITY BYTE ASSY FOR A READ FWD FUNCTION
2446                          ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2447                          ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2448
2449                          ;THIS IS PACKING MODE #2 - READ FORWARD
2450
2451  48FE  08                FD10C:  .BYTE   @10     ;1ST SCLK       DDR A (3 COMPLETE 36. BIT)
2452  48FF  04                        .BYTE   @4      ;               DDR B (WORDS OR 12 BYTES IN THE)
2453  4900  00                        .BYTE   @0      ;               DDR C (XFR. DATA = 1 TO 14(8))
2454  4901  00                        .BYTE   @0      ;2ND SCLK       DDR A
2455  4902  C1                        .BYTE   @301    ;               DDR B
2456  4903  00                        .BYTE   @0      ;               DDR C
2457  4904  18                        .BYTE   @30     ;3RD SCLK
2458  4905  14                        .BYTE   @24
2459  4906  00                        .BYTE   @0
2460  4907  00                        .BYTE   @0      ;4TH SCLK
2461  4908  C2                        .BYTE   @302
```

```
2462  4909  01                        .BYTE   @1
2463  490A  28                        .BYTE   @50     ;5TH SCLK
2464  490B  24                        .BYTE   @44
2465  490C  C0                        .BYTE   0
2466  490D  00                        .BYTE   @0      ;6TH SCLK (LAST)
2467  490E  C3                        .BYTE   @303
2468  490F  02                        .BYTE   @2
2469
2470                                ;TABLE OF DATA FOR A REVERSE READ DATA TRANSFER
2471
2472  4910  40                    RD10C:  .BYTE   @100    ;1ST SCLK       DDR A (3 COMPLETE 36. BIT)
2473  4911  80                            .BYTE   @200    ;               DDR B (WORDS OR 12 BYTES IN THE)
2474  4912  00                            .BYTE   @0      ;               DDR C (XFR. DATA = 14 TO 1(8))
2475  4913  0C                            .BYTE   @14     ;2ND SCLK       DDR A
2476  4914  10                            .BYTE   @20     ;               DDR B
2477  4915  00                            .BYTE   @0      ;               DDR C
2478  4916  40                            .BYTE   @100    ;3RD SCLK
2479  4917  81                            .BYTE   @201
2480  4918  01                            .BYTE   @1
2481  4919  1C                            .BYTE   @34     ;4TH SCLK
2482  491A  20                            .BYTE   @40
2483  491B  00                            .BYTE   @0
2484  491C  40                            .BYTE   @100    ;5TH SCLK
2485  491D  82                            .BYTE   @202
2486  491E  02                            .BYTE   2
2487  491F  2C                            .BYTE   @54     ;6TH SCLK (LAST)
2488  4920  30                            .BYTE   @60
2489  4921  00                            .BYTE   @0
2490                                ;TABLE OF EXPECTED DATA FOR 10-DUMP BYTE ASSY FOR A READ FWD FUNCTION
2491                                ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2492                                ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2493
2494                                ;THIS IS PACKING MODE #3 - READ FORWARD
2495
2496  4922  08                    FD10D:  .BYTE   @10     ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2497  4923  04                            .BYTE   @4      ;               DDR B (WORDS OR 10 BYTES IN THE)
2498  4924  00                            .BYTE   @0      ;               DDR C (XFR. DATA = 1 TO 12(8))
2499  4925  14                            .BYTE   @24     ;2ND SCLK       DDR A
2500  4926  C1                            .BYTE   @301    ;               DDR B
2501  4927  00                            .BYTE   @0      ;               DDR C
2502  4928  1C                            .BYTE   @34     ;3RD SCLK
2503  4929  18                            .BYTE   @30
2504  492A  00                            .BYTE   @0
2505  492B  68                            .BYTE   @150    ;4TH SCLK
2506  492C  02                            .BYTE   @2
2507  492D  02                            .BYTE   @2
2508
2509                                ;READ REVERSE TRANSFER EXPECTED DATA TABLE
2510
2511  492E  84                    RD10D:  .BYTE   @204    ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2512  492F  C0                            .BYTE   @300    ;               DDR B (WORDS OR 10 BYTES IN THE)
2513  4930  00                            .BYTE   @0      ;               DDR C (XFR. DATA = 12 TO 1(8))
2514  4931  10                            .BYTE   @20     ;2ND SCLK       DDR A
2515  4932  14                            .BYTE   @24     ;               DDR B
```

```
2516  4933  00                        .BYTE   @0      ;                       DDR C
2517  4934  D8                        .BYTE   @330    ;3RD SCLK
2518  4935  01                        .BYTE   @1
2519  4936  02                        .BYTE   @2
2520  4937  24                        .BYTE   @44     ;4TH SCLK
2521  4938  28                        .BYTE   @50
2522  4939  00                        .BYTE   @0
2523                          ;TABLE OF EXPECTED DATA FOR 10-HIGH DENSITY BYTE ASSY FOR A READ FWD FUNCTION
2524                          ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO RFPRESENT THE
2525                          ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2526
2527                          ;THIS IS PACKING MODE #4 (HDC) - READ FORWARD
2528                          ;ALSO PACKING MODE #6 (HDD) - READ FORWARD
2529
2530  493A                    FD10HC:
2531  493A  08                FD10HD: .BYTE   @10     ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2532  493B  04                        .BYTE   @4      ;               DDR B (WORDS OR 9 DATA BYTES)
2533  493C  00                        .BYTE   @0      ;               DDR C (IN XFR. DATA   1 THRU)
2534  493D  00                        .BYTE   @0      ;2ND SCLK       DDR A (11(8))
2535  493E  C1                        .BYTE   @301    ;               DDR B
2536  493F  00                        .BYTE   @0      ;               DDR C
2537  4940  81                        .BYTE   @201    ;3RD SCLK
2538  4941  41                        .BYTE   @101
2539  4942  01                        .BYTE   @1
2540  4943  27                        .BYTE   @47     ;4TH & LAST SCLK
2541  4944  20                        .BYTE   @40
2542  4945  00                        .BYTE   @0
2543
2544                          ;THIS IS PACKING MODE #4 (HDC) - READ REVERSE TABLE OF EXPECTED DATA
2545                          ;ALSO PACKING MODE #6 (HDD) - READ REVERSE DATA TABLE
2546
2547  4946                    RD10HC:
2548  4946  07                RD10HD: .BYTE   @7      ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2549  4947  08                        .BYTE   @10     ;               DDR B (WORDS OR 9 DATA BYTES)
2550  4948  00                        .BYTE   @0      ;               DDR C (IN XFR. DATA   11 THRU)
2551  4949  0C                        .BYTE   @0      ;2ND SCLK       DDR A (1(8))
2552  494A  41                        .BYTE   @101    ;               DDR B
2553  494B  01                        .BYTE   @1      ;               DDR C
2554  494C  80                        .BYTE   @200    ;3RD SCLK
2555  494D  C1                        .BYTE   @301
2556  494E  01                        .BYTE   @1
2557  494F  20                        .BYTE   @40     ;4TH & LAST SCLK
2558  4950  24                        .BYTE   @44
2559  4951  00                        .BYTE   @0
2560                          ;TABLE OF EXPECTED DATA FOR IMAGE BYTE ASSY FOR A READ FWD FUNCTION
2561                          ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2562                          ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2563
2564                          ;THIS IS PACKING MODE #5 - READ FORWARD AND READ REVERSE
2565
2566  4952                    RDIMAG:
2567  4952  04                FDIMAG: .BYTE   @4      ;1ST SCLK       DDR A
2568  4953  00                        .BYTE   @0      ;               DDR B
2569  4954  00                        .BYTE   @0      ;               DDR C
```

```
2570  4955  08          .BYTE   @10     ;SCLK #2        DDR A
2571  4956  00          .BYTE   @0      ;               DDR B
2572  4957  00          .BYTE   @0      ;               DDR C
2573  4958  0C          .BYTE   @14     ;SCLK #3        DDR A
2574  4959  00          .BYTE   @0      ;               DDR B
2575  495A  00          .BYTE   @0      ;               DDR C
2576  495B  10          .BYTE   @20     ;SCLK #4        DDR A
2577  495C  00          .BYTE   @0      ;               DDR B
2578  495D  00          .BYTE   @0      ;               DDR C
2579  495E  14          .BYTE   @24     ;SCLK #5        DDR A
2580  495F  00          .BYTE   @0      ;               DDR B
2581  4960  00          .BYTE   @0      ;               DDR C
2582  4961  18          .BYTE   @30     ;SCLK #6        DDR A
2583  4962  00          .BYTE   @0      ;               DDR B
2584  4963  00          .BYTE   @0      ;               DDR C
2585  4964  1C          .BYTE   @34     ;SCLK #7        DDR A
2586  4965  00          .BYTE   @0      ;               DDR B
2587  4966  00          .BYTE   @0      ;               DDR C
2588  4967  20          .BYTE   @40     ;SCLK #10       DDR A
2589  4968  00          .BYTE   @0      ;               DDR B
2590  4969  00          .BYTE   @0      ;               DDR C
2591  496A  24          .BYTE   @44     ;SCLK #11       DDR A
2592  496B  00          .BYTE   @0      ;               DDR B
2593  496C  00          .BYTE   @0      ;               DDR C
2594  496D  28          .BYTE   @50     ;SCLK #12       DDR A
2595  496E  00          .BYTE   @0      ;               DDR B
2596  496F  00          .BYTE   @0      ;               DDR C
2597  4970  2C          .BYTE   @54     ;SCLK #13       DDR A
2598  4971  00          .BYTE   @0      ;               DDR B
2599  4972  00          .BYTE   @0      ;               DDR C
2600  4973  30          .BYTE   @60     ;SCLK #14       DDR A
2601  4974  00          .BYTE   @0      ;               DDR B
2602  4975  00          .BYTE   @0      ;               DDR C
2603  4976  34          .BYTE   @64     ;SCLK #15       DDR A
2604  4977  00          .BYTE   @0      ;               DDR B
2605  4978  00          .BYTE   @0      ;               DDR C
2606  4979  38          .BYTE   @70     ;SCLK #16       DDR A
2607  497A  00          .BYTE   @0      ;               DDR B
2608  497B  00          .BYTE   @0      ;               DDR C
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS

                                SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CLEAR ALL TU PORTS
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*     CLEAR PORT SELECT FOR TRANSPORT
                                ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                ;*     CLEAR PORT AMTIE WORD
                                ;*ENDSUB
                                S
                                ; ***************************************************************************
 497C  F5                12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
 497D  C5                12.0          PUSH    B
 497E  06  00             7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
 4980  DB  E0            10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
 4982  E6  80             7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
 4984  B0                4.0          ORA     B               ;ADD IN THE SELECTED PORT #
 4985  D3  E0            10.0          OUT     MBSEL           ;RESET TO THIS PORT
 4987  3E  80             7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
 4989  D3  40            10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
 498B  AF                4.0          XRA     A               ;CLEAR TU COMMAND A
 498C  D3  40            10.0          OUT     TCMD            ;
 498E  3E  81             7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
 4990  D3  40            10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
 4992  3E  00             7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
 4994  D3  40            10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
 4996  AF                4.0          XRA     A
 4997  D3  44            10.0          OUT     TAMT            ;CLEAR AMTIE WORD
 4999  D3  48            10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
 499B  D3  4C            10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
 499D  04                4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
 499E  78                4.0          MOV     A,B
 499F  FE  04            7.0          CPI     4               ;DONE?
 49A1  C2  80  49       10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
 49A4  C1               10.0          POP     B               ;RESET B & C
 49A5  F1               10.0          POP     PSW             ;ALL DONE
 49A6  C9               10.0          RET                     ;EXIT
```

```
                                  .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
     49A7                         SSUB
 (1)                              ; *******************************************************************
 (1)                              ;*SUBROUTINE TITLE
 (1)                              ;*---------- -----
                                  ;*4X5 TRANSLATE A SUBGROUP
     49A7                         SD
 (1)                              ; *******************************************************************
 (1)                              ;*DESCRIPTION
 (1)                              ;*-----------
                                  ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                  ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
     49A7                         SP
 (1)                              ; *******************************************************************
 (1)                              ;*PROCEDURE
 (1)                              ;*---------
                                  ;*BGNSUB
                                  ;*      GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                  ;*      CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                  ;*      SET LOOP COUNT TO 9
                                  ;*      BGNDO
                                  ;*      :     CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                  ;*      :     ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                  ;*      :     GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                  ;*      :         FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                  ;*      :         BYTE OF THE INPUT DATA, ETC.)
                                  ;*      :     REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                  ;*      :         TO BUILD THE INDEX)
                                  ;*      :     USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                  ;*      :     PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                  ;*      :         BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                  ;*      :         TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                  ;*      :         DATA BYTE.
                                  ;*      :     DECREMENT THE LOOP COUNT
                                  ;*      :     DO UNTIL THE LOOP COUNT   0
                                  ;*      ENDDO
                                  ;*ENDSUB
     49A7                         S
 (1)                              ; *********************************************************************

     49A7  F5            12.0  T4X5:  PUSH   PSW           ;SAVE REGISTER A + PSW
     49A8  C5            12.0         PUSH   B             ;SAVE REGISTER B + C
     49A9  D5            12.0         PUSH   D             ;SAVE REGISTER D + E
     49AA  E5            12.0         PUSH   H             ;SAVE REGISTER H + L
                               ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                               ;CLEAR THE TRANSLATED DATA TABLE.
     49AB  06 0A          7.0         MVI    B,10          ;SET UP LOOP COUNT
     49AD  11 35 4A      10.0         LXI    D,TRNOUT      ;GET POINTER TO TRANSLATED DATA TABLE
     49B0  AF             4.0         XRA    A             ;CLEAR A
     49B1  12             7.0  D4X5:  STAX   D             ;STORE IN THE TABLE
     49B2  13             6.0         INX    D             ;UPDATE TABLE POINTER
     49B3  05             4.0         DCR    B             ;DECREMENT LOOP COUNT
     49B4  C2 B1 49      10.0         JNZ    D4X5          ;DO UNTIL LOOP COUNT-0
     49B7  0E 09          7.0         MVI    C,9           ;SET UP TRACK COUNT
```

```
2699  49B9  06  04        7.0  B4X5:  MVI   B,4          ;SET UP BIT COUNT
2700  49BB  11  31  4A    10.0        LXI   D,TRNIN
2701  49BE  AF             4.0        XRA   A            ;CLEAR THE GROUP POSITION COUNT
2702  49BF  32  3F  4A    13.0        STA   GP4X5
2703  49C2  1A             7.0  C4X5:  LDAX  D            ;GET A DATA BYTE
2704  49C3  A7             4.0        ANA   A            ;SET CONDITION BITS
2705  49C4  E2  C8  49    10.0        JPO   PO4X5        ;ODD PARITY LEAVE CARRY CLEAR
2706  49C7  37             4.0        STC                ;EVEN PARITY SET CARRY
2707  49C8  1F             4.0  PO4X5: RAR              ;SHIFT OUT DESIRED BIT
2708  49C9  12             7.0        STAX  D            ;STORE RESULT BACK IN TEMP TABLE
2709  49CA  3A  3F  4A    13.0        LDA   GP4X5        ;GET THE GROUP POSITION BYTE
2710  49CD  17             4.0        RAL                ;PUT IN THIS DATA BIT
2711  49CE  32  3F  4A    13.0        STA   GP4X5        ;SAVE THE UPDATED GROUP POSITION BYTE
2712  49D1  13             6.0        INX   D            ;UPDATE THE TABLE POINTER
2713  49D2  05             4.0        DCR   B            ;DECREMENT THE BIT COUNT
2714  49D3  C2  C2  49    10.0        JNZ   C4X5         ;DO UNTIL ALL 4 BITS ARE RECEIVED
2715
2716                                ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2717
2718                                ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2719  49D6  06  05        7.0        MVI   B,5          ;SET UP LOOP COUNT
2720  49D8  11  35  4A    10.0        LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2721  49DB  13             6.0  E4X5:  INX   D            ;POINT TO PARITY BIT
2722  49DC  1A             7.0        LDAX  D            ;GET PARITY BIT
2723  49DD  A7             4.0        ANA   A            ;CLEAR THE CARRY BIT
2724  49DE  1F             4.0        RAR                ;SHIFT TO THE CARRY BIT
2725  49DF  12             7.0        STAX  D            ;STORE IT BACK
2726  49E0  1B             6.0        DCX   D            ;DECREMENT TO DATA BITS
2727  49E1  1A             7.0        LDAX  D            ;GET THE DATA BITS
2728  49E2  1F             4.0        RAR                ;SHIFT IN CARRY BITS
2729  49E3  12             7.0        STAX  D            ;STORE BACK IN TABLE
2730  49E4  13             6.0        INX   D            ;POINT TO NEXT TABLE ENTRY
2731  49E5  13             6.0        INX   D            ;
2732  49E6  05             4.0        DCR   B            ;DECREMENT LOOP COUNT
2733  49E7  C2  DB  49    10.0        JNZ   E4X5         ;DO UNTIL LOOP COUNT-0
2734  49EA  11  17  4A    10.0        LXI   D,TAB4X5     ;LOAD ADDRESS OF TRANSLATION TABLE
2735  49ED  26  00         7.0        MVI   H,0          ;CLEAR REGISTER H
2736  49EF  3A  3F  4A    13.0        LDA   GP4X5        ;GET GROUP POSITION COUNT
2737  49F2  6F             4.0        MOV   L,A
2738  49F3  19            10.0        DAD   D            ;ADD GROUP POSITION COUNT TO TABLE
2739  49F4  7E             7.0        MOV   A,M          ;GET TRANSLATED DATA
2740  49F5  32  40  4A    13.0        STA   TRNTMP
2741                                ;''TRNTMP'' = THE TRANSLATED DATA
2742
2743  49F8  11  35  4A    10.0        LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2744  49FB  06  05         7.0        MVI   B,5          ;SET UP LOOP COUNT
2745  49FD  3A  40  4A    13.0  F4X5:  LDA   TRNTMP      ;GET TRANSLATED DATA
2746  4A00  13             6.0        INX   D            ;POINT TO PARITY POSITION
2747  4A01  A7             4.0        ANA   A            ;CLEAR CARRY BIT
2748  4A02  1F             4.0        RAR                ;SHIFT OUT A BIT
2749  4A03  32  40  4A    13.0        STA   TRNTMP       ;STORE TRANSLATED DATA
2750  4A06  1A             7.0        LDAX  D            ;GET THE PARITY BIT (ALWAYS ZERO)
2751  4A07  17             4.0        RAL                ;ROLL IN THE CARRY BIT
2752  4A08  12             7.0        STAX  D            ;STORE AS THE PARITY BIT
```

SEQ 1113

```
2753  4A09  13                  6.0          INX     D              ;UPDATE TABLE POINTER
2754  4A0A  05                  4.0          DCR     B              ;DECREMENT THE LOOP COUNT
2755  4A0B  C2  FD  49          10.0         JNZ     F4X5           ;DO UNTIL LOOP COUNT=0
2756
2757                                    ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2758
2759  4A0E  0D                  4.0          DCR     C              ;DECREMENT THE TRACK COUNT
2760  4A0F  C2  B9  49          10.0         JNZ     B4X5           ;DO UNTIL ALL TRACKS TRANSLATED
2761  4A12  E1                  10.0         POP     H              ;RESTORE REGISTER H + L
2762  4A13  D1                  10.0         POP     D              ;RESTORE REGISTER D + E
2763  4A14  C1                  10.0         POP     B              ;RESTORE REGISTER B + C
2764  4A15  F1                  10.0         POP     PSW            ;RESTORE REGISTER A + PSW
2765  4A16  C9                  10.0         RET                    ;RETURN TO USER
```

```
2767                                    .SBTTL  TABLE 4 X 5 TRANSLATION
2768
2769                                    ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2770                                    ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2771                                    ;   1 2 3 4 / 5 6 7 8                  10 9 8 7 6 / 5 4 3 2 1
2772                                    ;
2773                                    ;       0000                              10011
2774                                    ;       0001                              11011
2775                                    ;       0010                              01001
2776                                    ;       0011                              11001
2777                                    ;       0100                              10111
2778                                    ;       0101                              10101
2779                                    ;       0110                              01101
2780                                    ;       0111                              11101
2781                                    ;       1000                              01011
2782                                    ;       1001                              10010
2783                                    ;       1010                              01010
2784                                    ;       1011                              11010
2785                                    ;       1100                              01111
2786                                    ;       1101                              10110
2787                                    ;       1110                              01110
2788                                    ;       1111                              11110
2789
2790  4A17  13                          TAB4X5: .BYTE   @23
2791  4A18  1B                                  .BYTE   @33
2792  4A19  09                                  .BYTE   @11
2793  4A1A  19                                  .BYTE   @31
2794  4A1B  17                                  .BYTE   @27
2795  4A1C  15                                  .BYTE   @25
2796  4A1D  0D                                  .BYTE   @15
2797  4A1E  1D                                  .BYTE   @35
2798  4A1F  0B                                  .BYTE   @13
2799  4A20  12                                  .BYTE   @22
2800  4A21  0A                                  .BYTE   @12
2801  4A22  1A                                  .BYTE   @32
2802  4A23  0F                                  .BYTE   @17
2803  4A24  16                                  .BYTE   @26
2804  4A25  0E                                  .BYTE   @16
2805  4A26  1E                                  .BYTE   @36
```

```
2807                                    .SBTTL  SUBROJTINE VARIABLES
2808
2809  4A27  00                          SDDRA:  .BYTE   0               ;STORAGE FOR DDR A DATA
2810  4A28  00                          SDDRB:  .BYTE   0               ;STORAGE FOR DDR B DATA
2811  4A29  00                          SDDRC:  .BYTE   0               ;STORAGE FOR DDR C DATA
2812  4A2A  00                          SCLKCT: .BYTE   0               ;COUNT OF 'SCLK' RECEIVED
2813  4A2B  00                          ESCLK:  .BYTE   0               ;NUMBER OF EXPECTED SCLK PER PASS
2814  4A2C  00                          FMTNUM: .BYTE   0               ;SLFECTED PACKING MODE PER PASS
2815  4A2D  00                          UNITMP: .BYTE   0               ;PORT NUMBER TO USE AS RESPONDED BY HOST
2816  4A2E  00                          SUNIT:  .BYTE   0               ;STORAGE OF THE SELECTED UNIT NUMBER
2817  4A2F  00                          REVERSE: .BYTE  0               ;READ FWD/REV FLAG ... -0 FOR READ FWD
2818  4A30  00                          CCTLWD: .BYTE   0               ;LOAD INTO THE SYSTEM CLOCK CONTROL WORD
2819
2820  4A31  0004                        TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2821                                                                    ;BE TRANSLATED
2822  4A35  000A                        TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2823                                                                    ;TRANSLATION
2824  4A3F  00                          GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2825                                                                    ;BE TRANSLATED
2826  4A40  00                          TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2827
2828  4A41  00                          MOREFG: .BYTE   0               ;FLAG FOR LOADING AN ADDITIONAL DATA GRP
```

```
                                        .SBTTL  SUBROUTINE CLEAR ECC
2830
2831  4A42                              S
 (1)                                    ; ************************************************************************
2832                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
2833                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2834                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2835  4A42                              S
 (1)                                    ; ************************************************************************
2836
2837  4A42  AF                    4.0   CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
2838  4A43  32   6E   4A         13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
2839  4A46  C9                   10.0          RET                     ;RETURN TO USER
```

```
2841                                    .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
2842  4A47                              $
 (1)                                    ; ************************************************************************
2843                                    ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
2844                                    ;THE CONTENTS OF ''ECCCHR'' TO UPDATE THE CONTENTS OF ''ECCCHR'' ACCORDING
2845                                    ;TO THE ANSI STANDARD ECC POLYNOMIAL.
2846  4A47                              $
 (1)                                    ; ************************************************************************
2847
2848  4A47  E5              12.0  ECC:    PUSH    H
2849  4A48  C5              12.0          PUSH    B
2850  4A49  21  6E  4A      10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
2851  4A4C  AE               7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
2852  4A4D  5F               4.0          MOV     E,A             ;SAVE XOR RESULT IN E
2853  4A4E  E6  10           7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
2854  4A50  7B               4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
2855  4A51  CA  56  4A      10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
2856  4A54  EE  23           7.0          XRI     $23             ;ELSE-XOR WITH 23
2857  4A56  5F               4.0  ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
2858  4A57  AF               4.0          XRA     A               ;CLEAR A
2859  4A58  4F               4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
2860  4A59  21  66  4A      10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
2861  4A5C  CD  6F  4A      18.0          CALL    TRANS           ;TRANSLATE THE BITS
2862  4A5F  79               4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
2863  4A60  32  6E  4A      13.0          STA     ECCCHR          ;STORE RESULT
2864  4A63  C1              10.0          POP     B
2865  4A64  E1              10.0          POP     H
2866  4A65  C9              10.0          RET
2867
2868  4A66  08                    ECCTBL: $08                     ;BIT 0 = POSITION 3
2869  4A67  20                            $20                     ;BIT 1 = POSITION 5
2870  4A68  02                            $02                     ;BIT 2 = POSITION 1
2871  4A69  40                            $40                     ;BIT 3 = POSITION 6
2872  4A6A  80                            $80                     ;BIT 4 = POSITION 7
2873  4A6B  01                            $01                     ;BIT 5 = POSITION 0
2874  4A6C  10                            $10                     ;BIT 6 = POSITION 4
2875  4A6D  04                            $04                     ;BIT 7 = POSITION 2
2876
2877  4A6E  00                    ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;
4A6F  06  01            7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
4A71  7B                4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
4A72  A0                4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
4A73  CA  79  4A       10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
4A76  79                4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
4A77  B6                7.0          ORA   M            ;'OR' IN NEW POSITION
4A78  4F                4.0          MOV   C,A          ;SAVE RESULT

4A79  78                4.0  TRANS2: MOV   A,B
4A7A  07                4.0          RLC                ;POSITION MASK TO NEXT BIT
4A7B  47                4.0          MOV   B,A
4A7C  D8               12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
4A7D  23                6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
4A7E  C3  71  4A       10.0          JMP   TRANS1       ;PROCESS NEXT BIT

4A81                                S
 (1)                                ; *****************************************************************************
                                    ;CLKSYS - ROUTINE TO CLOCK THE SYSTEM USING THE CONTENTS OF 'CCTLWD'
                                    ;       AS A BASE AND TOGGLING THE CLOCK BIT IN THE CLOCK CONTROL WORD
4A81                                S
 (1)                                ; *****************************************************************************

4A81  F5               12.0  CLKSYS: PUSH  PSW
4A82  3A  30  4A       13.0          LDA   CCTLWD       ;GET THE CONTROL WORD BASE
4A85  F6  40            7.0          ORI   SSCLK        ;GET THE CLOCK BIT
4A87  D3  F0           10.0          OUT   CLKCTL       ;SET THE CLOCK ON
4A89  E6  BF            7.0          ANI   @277         ;CLEAR THE CLOCK BIT
4A8B  D3  F0           10.0          OUT   CLKCTL       ;FINISH THE CLOCK CYCLE
4A8D  F1               10.0          POP   PSW
4A8E  C9               10.0          RET

      0000                      .END
```

```
A      =%0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
ARAIDF= 0098    ASAVE   4F9B    ATTCD   4F97    AXNUM   4F91
B      =%0000   BADST  = 0090   BIT0   = 0001   BIT1   = 0002
BIT15  = 8000   BIT2   = 0004   BIT3   = 0008   BIT4   = 0010
BIT5   = 0020   BIT6   = 0040   BIT7   = 0080   BIT8   = 0100
BIT9   = 0200   BRKPBC= 4F0A    BRKRAM= 4F10    BRKSTR= 4E60
BRKXCT= 4F00    BSAVE   4F9C    BYTCNT= 00D4    BYTEH   4F24
BYTEL   4F23    B4X5    49B9    C      =%0001   CASCT   4F21
CASCTL= 00A0    CASSTA= 00A0    CATTH  = 0089   CATTL  = 0088
CBUSST= 00A1    CBYTH  = 008B   CBYTL  = 008A   CCTLWD  4A30
CDG1H  = 0087   CDG1L  = 0086   CDG2H  = 0093   CDG2L  = 0092
CDG3H  = 0095   CDG3L  = 0094   CDVTH  = 008D   CDVTL  = 008C
CHPTIE= 0028    CH0TIE= 0020    CH1TIE= 0021    CH2TIE= 0022
CH3TIE= 0023    CH4TIE= 0024    CH5TIE= 0025    CH6TIE= 0026
CH7TIE= 0027    CKLOP  = 2815   CLEAR   497C    CLECC   4A42
CLKCTL= 00F0    CLKRS   43E6    CLKRS2  462C    CLKSYS  4A81
CLOCK   4F26    CLRLP   4980    CMC0H  = 0099   CMC0L  = 0098
CMC1H  = 009B   CMC1L  = 009A   CMC2H  = 009D   CMC2L  = 009C
CMC3H  = 009F   CMC3L  = 009E   CMINH  = 0097   CMINL  = 0096
CNTCTL= C0D7    CNV1    4849    CONVRT  4843    CRCG    467F
CRCGEC  4686    CRCG2   4683    CRCWRD= 0018    CSAVE   4F9D
CSRLH  = 0091   CSRLL  = 0090   CTCH   = 0085   CTCL   = 0084
CTSTH  = 008F   CTSTL  = 008E   CXCTH  = 0081   CXCTL  = 0080
CXINH  = 0083   CXINL  = 0082   C.     = 0001   C.AVAI= 0080
C.DP   = 0008   C.DSE  = 0010   C.DTU  = 0003   C.DVA  = 0008
C.FAIL= 00FC    C.FMT  = 0070   C.FNCT= 003E    C.GO   = 0001
C.INTC= 00FE    C.MAIN= 0020    C.NSA  = 0080   C.RCT  = 00FC
C.SER  = 0080   C.SHR  = 0040   C.SKPC= 000F    C.TAPE= 0040
C.WCS  = 0002   C1EC1   43F1    C1EC1L  43F9    C1EC2   4408
C1EC2L  4410    C2EC1   4637    C2EC1L  463F    C2EC2   464E
C2EC2L  4656    C2EC3   4665    C2EC3L  466D    C4X5    49C2
D      =%0002   DATACT= 00D0    DBUS    4F28    DBUSCT= 00C0
DBUSST= 00C0    DDRA   = 00D8   DDRAIN= 0010    DDRB   = 00D9
DDRBIN= 0002    DDRC   = 00DA   DDRCIN= 0001    DDRCO  = 0088
DDRCTL= 00DB    DG1     488F    DG1ECC  4896    DG2     4897
DG2ECC  489E    DIAFLG  4F22    DIAGPG= 4300    DIAGRM= 4F90
DIARD  = 000B   DONE1  = 0045   DONINT= 0010    DSAVE   4F9E
DSE    = 0006   DUMM1   447E  G  DUMM2   449E  G  DUMM3   4722  G
DUMM4   4758  G  D.ATH0= 0001   D.ATH1= 0002    D.EOTD= 0010
D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004    D.TACH= 0008
D.WR4  = 0080   D4X5    49B1    E      =%0003   ECC     4A47
ECCBAD= 0042    ECCCHR  4A6E    ECCCOR= 0019    ECCOK  = 0041
ECCSTA= 001A    ECCTBL  4A66    ECCTST= 000E    ECC1    4A56
EDATA  = 0095   EDFWD   48AF    EDREV   48BD    EOTCLR= 0003
ERFLG   4F93    ERLP   = 2809   ERLPA  = 280F   ERLPB  = 2812
ERLPE  = 280C   ERNUM   4F90    ERRCNT= 00D6    ESAVE   4F9F
ESCLK   4A2B    ESCLKT  48CB    EXIT    456F    EXIT2   4829
EXPADR  48D2    E.ACRC= 0010    E.AMT  = 0020   E.CDP  = 0080
E.CRC  = 0080   E.PNTR= 0008    E.RPE  = 0040   E.STEC= 0001
F.TTEC= 0002    E.UNC  = 0004   E4X5    49DB    FDIMAG  4952
FD10C   48FE    FD10D   4922    FD10HC  493A    FD10HD  493A
FD11N   48D4    FD15N   48E9    FIFORD= 006A    FLOAD   485F
FMTNUM  4A2C    FORMAT  4F25    FOUND   4327    FPORT   4347
FWDTST= 0061    F4X5    49FD    GCRID  = 0089   GCRSET= 0002
```

```
GOODTM= 0092        GP4X5   4A3F        H      =%0004      HLSAVE  4FA0
IE     = 0008       INTSTA= 00E0        ITERA   4F9A       I.PWR  = 0020
I.RMPE= 0040        I5.5   = 0010       I6.5   = 0020       I7.5   = 0040
KCALL  = 005F       KCLR   = 007B       KDEP   = 003F       KENAB  = 0078
KEXAM  = 003E       KEYBRD= 00C8        KEY1   = 0078       KEY10  = 006D
KEY11  - 006E       KEY12  = 006F       KEY13  = 005C       KEY14  = 005D
KEY15  = 005E       KEY16  = 005F       KEY17  = 003C       KEY18  = 003D
KEY19  = 003E       KEY2   = 0079       KEY20  = 003F       KEY3   = 007A
KEY4   = 007B       KEY5   = 0074       KEY6   = 0075       KEY7   = 0076
KEY8     0077       KEY9   = 006C       KINTA  = 006F       KLDAD  = 003D
KN0    = 003C       KN1    = 005C       KN2    = 005D       KN3    = 005E
KN4    = 006C       KN5    = 006D       KN6    = 006E       KN7    = 0074
KN8    - 0075       KN9    = 0076       KU2    = 0079       KU3    = 007A
KU8    = 0077       L      =%0005       LBLANK= 000F        LCE    = 000B
LCH    = 000C       LCL    = 000D       LCP    = 000E       LC0    = 0000
LC1    = 0001       LC2    = 0002       LC3    = 0003       LC4    = 0004
LC5    = 0005       LC6    = 0006       LC7    = 0007       LC8    = 0008
LC9    - 0009       LDFIFO  4864        IDLEDA= 00CA        LDLEDB= 00CB
LDLEDC= 00CC        LDLEDD= 00CD        LDLEDE- 00CE        LDLEDF= 00CF
LKDIAG= 2800        LKKBD  = 004C       LKKEY  = 0049       LKLWMG= 0058
LKLWMP= 0055        LKLWPG= 0052        LKLWPP= 004F        LKMOD7= 0046
LKOPR  = 0046       LOADIT  4858        LPFLG   4F94        LPNUM   4F92
M      -%0006       MBSEL  = 00E0       MB.A   = 0008       MB.B   = 0004
MEMTOP- 4FFF        MINUS  = 000A       MM     = 8000       MOREFG  4A41
MSE    = 0008       MSGN   = 000B       MTACLR= 0000        MT.ARA= 0020
MT.CPE= 0080        MT.DSE= 0001        MT.FWD= 0040        MT.INH= 0008
MT.LWR= 0004        MT.MOT= 0002        MT.NWT= 0080        MT.PEC= 0040
MT.PSB- 0004        MT.PS0= 0001        MT.PS1= 0002        MT.REV= 0020
MT.WRT- 0010        MT.Z   - 0008       M.ATA  = 0080       M.CAPE= 0020
M.CONT= 0080        M.DEM  = 0020       M.EBL  = 0010       M.EXC  = 0008
M.FAIL= 0008        M.ILR  = 0010       M.INIT= 0010        M.OCC  = 0020
M.ONLI- 0001        M.PE   = 0040       M.PORT= 0080        M.RDEN= 0002
M.RDPE- 0008        M.RUN  = 0004       M.SCLK= 0001        M.TRA  = 0040
M.UNIT- 0007        M.WCLK= 0040        M.WCLN= 0080        M.WREN= 0080
M5.5    0001        M6.5   = 0002       M7.5   = 0004       NOTCAP= 0088
NOUNIT  431E  G     OKAY   = 00FF       OPRRAM= 4300        OPSTRT= 0058
OPVER  = 0040       PADCNT= 00D5        PADCRC= 0080        PDIAG  = 0048
PEID   - 008A       PENAB  = 004C       PESET  = 0001       PL     = 00B1
PO4X5   49C8        PRDD   = 004C       PRENF  = 009C       PS     = 00B2
PSTAT   0048        PSW    =%0009       P.AMTP= 0001        P.BCTC= 0040
P.CMDP- 0020        P.INTE= 0080        P.LCS  = 0040       P.LWR  = 0020
P.RDP   0002        P.RPEN= 0004        P.RPST= 0002        P.RPOE- 0020
P.RP1E= 0010        P.RP2E= 0020        P.RP3E= 0010        P.SING= 0080
P.STAT= 0002        P.STPE- 0080        P.TACH= 0008        P.TUPR= 0010
P.WCSP- 0004        P.WDS  = 0040       P.WFLP= 0001        P.WPEN= 0010
P.WPOE- 0008        P.WP1E= 0004        P.WP2E= 0008        P.WP3E= 0004
P.5VOK- 0002        QUE    = 281B       QUEM   = 281E       RAMT   = 0010
RARA   = 0006       RARAI  = 0004       RCHBD0= 0048        RCHBD1= 0047
RCHOK  - 0046       RCHTST= 000C        RCLRT  = 000D       RCMD   = 000B
RCMLP  - 0003       RCONT  = 0080       RDATA  = 0017       RDCLK  = 0010
RDG1    489F        RDG1EC  48A6        RDG2    48A7        RDG2EC  48AE
RDIMAG  4952        RDON   = 0011       RD10C   4910        RD10D   492E
RD10HC  4946        RD10HD  4946        RD11N   48E9        RD15N   48D4
READG  - 0007       REND    0014        REQST  = 2806        RESCHR= 00D1
```

```
RESG    467F          RESG2   4683          REVERS  4A2F          REVTST= 0064
REWIND= 0004          RFIFOL= 0008          RGCLK = 0002          RGCRI = 0003
RIBG  = 0001          RILL  = 0012          RINST = 000C          RLFIFO  487B
RLOAD   4876          RMCTST= 0008          RMK2  = 0013          RNOP  = 0000
RPATH = 0001          RPBAD = 0044          RPCHI = 0001          RPCLK = 0003
RPCTL = 0009          RPEI  = 0002          RPFAIL= 0000          RPF1  = 009D
RPF2  = 009E          RPOK  = 0043          RPOSTN= 0016          RPSTA = 0015
RRCMT = 000A          RSG2    48A3          RSG4    48AB          RSTAT = 0002
RTIEB = 000A          RTIER = 0030          RTM   = 0005          RUNKI = 0009
RUPTST= 005E          RWDUNL= 0005          R.AMT = 0001          R.BOP = 0008
R.DATA= 0040          R.DON = 0002          R.DRDY= 0010          R.END = 0010
R.ILL = 0004          R.JVOK= 0004          R.MK2 = 0008          R.PLOD= 0008
R.PLOO= 0010          R.PL01= 0020          R.POST= 0020          R.STNM= 0002
R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040
R.VOK = 0080          R00H  = 0081          R00L  = 0080          R01H  = 0083
R01L  = 0082          R02H  = 0085          R02L  = 0084          R03H  = 0087
R03L  = 0086          R04H  = 0089          R04L  = 0088          R05H  = 008B
R05L  = 008A          R06H  = 008D          R06L  = 008C          R07H  = 008F
R07L  = 008E          R10H  = 0091          R10L  = 0090          R11H  = 0093
R11L  = 0092          R12H  = 0095          R12L  = 0094          R13H  = 0097
R13L  - 0096          R14H  = 0099          R14L  = 0098          R15H  = 009B
R15L  - 009A          R16H  = 009D          R16L  = 009C          R17H  = 009F
R17L  - 009E          R7.5  = 0010          SCLKCT  4A2A          SDDRA   4A27
SDDRB   4A28          SDDRC   4A29          SELCLR= 0000          SETATA= 00A1
SG2     4893          SG4     489B          SID   = 0080          SOD   = 0080
SOE   - 0040          SP    -%0008          SSCLK = 0040          SSTEP = 0005
STACK = 4FFF          STATRM= 4F20          STPCT   4F20          STRSP = 5000
SUNIT   4A2E          TAB4X5  4A17          TADR00= 0080          TADR01= 0081
TADR02- 0082          TADR03= 0083          TADR04= 0084          TADR05= 0085
TADR06- 0086          TADR07= 0087          TADR10= 0088          TADR11= 0089
TADR12= 008A          TADR13= 008B          TAMT  = 0044          TASEL = 0080
TCMD    0040          TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004
TC.REV= 0020          TC.WRT= 0010          TEMP    4F99          TEST1   4300
TEST2   4586          TMF   = 0099          TMRDY = 0040          TRANS   4A6F
TRANS1  4A71          TRANS2  4A79          TRKENA= 00D2          TRNIN   4A31
TRNOUT  4A35          TRNTMP  4A40          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0- 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002          T1CKA   44D1
T1CKAC  44FA  G       T1CKB   4500          T1CKBC  4514  G       T1CKC   451A
T1CKCC  452E  G       T1CLK   4469          T1CLKA  4463          T1LOOP  4370
T1PLO   441F          T1PM0   4568          T1PM6   455B          T1RF    4388
T1SCKL  4489          T1SCK0  44A7          T1SCLK  4487          T1SG1   4437
T1SG2   4440          T1SG3   4449          T1SG4   4452          T1ST    43EF
T2CKA   478B          T2CKAC  47B4  G       T2CKB   47BA          T2CKBC  47CE  G
T2CKC   47D4          T2CKCC  47E8  G       T2CLK   470D          T2CLKA  4707
T2EMK   46C1          T2EMKL  46C6          T2LOOP  45B6          T2LP    458B
T2PLO   4687          T2PM0   4822          T2PM6   4815          T2RR    45CE
T2SCKL  4743          T2SCK0  4761          T2SCLK  472B          T2SGD1  46DE
T2SGD2  46D5          T2SGD3  46F0          T2SGD4  46E7          T2SG1   469D
T2SG2   46A6          T2SG3   46AF          T2SG4   46B8          T2ST    4635
T4X5    49A7          UIBG  = 00A1          UNITMP  4A2D          VALFC   4F98
```

```
VALTB   4F95            VELTST= 005B            WDR.P - 0010            WMCCTL= 00D3
WMCERR= 00DA            WMCSTA= 00D0            WRTCLK= 0000            WRTDAT= 00D3
W.ACRC= 0004            W.CRC = 0008            W.DIAG= 0002            W.DONN= 0040
W.ECC = 0010            W.ENAB= 0080            W.ERR = 0020            W.FMT = 0070
W.GCR = 0010            W.LEFT= 0004            W.ONES= 0020            W.RESI= 0002
W.REV = 0004            W.ROME= 0010            W.RST = 0001            W.SKIP= 000F
W.WRIT= 0008            W.XFER= 0020            X     =%000A            X.DONN= 0080
X.ENAB= 0040            X.PEPE= 0002            X.ROME- 0001            X.WCLK= 0001
Y     =%000B            .     = 4A8F


 ERRORS DETECTED:  0

 *WMC6.A78/PTP,WMC6=NLIST,PARAM,MACRO,LIST,WMC6
 RUN-TIME: 5 7 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:08 SEQ 1123
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  WMC7 - BYTE ASSEMBLY/ZERO FILL OF READ PATH DATA
                                .SBTTL  TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
                                ;ID     WMC7-WRITE MICRO CONTROLLER PART #7
1340  4300                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1341                            ;*WMC BYTE ASSEMBLY/ZERO FILL TEST FROM READ FORWARD DATA
1342  4300                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1343                            ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
1344                            ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
1345                            ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
1346                            ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
1347                            ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
1348                            ;*TO 10-HI DEN COMPAT) IN A READ FORWARD DATA TRANSFER. THE TEST WILL
1349                            ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
1350                            ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
1351                            ;*EXPECTED. THE READ PATH WILL BE FILLED WITH DATA TO SIMULATE SHORT DATA
1352                            ;*DATA TRANSFERS (ONLY RESIDUAL DATA GROUPS - XFR LENGTH FROM 1 - 6 BYTES).
1353                            ;*THIS TEST WILL VERIFY THAT THE WMC CAN SEND THE PROPER AMOUNT OF 'PAD'
1354                            ;*BYTES TO FILL AN 'SCLK' TRANSFER OF 18 BITS DATA TO THE HOST IN THE
1355                            ;*SELECTED DATA PACKING MODE.
1356  4300                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1357                            ;*BGNTST
1358                            ;*  REQUEST A TU PORT FROM 'HOST' CPU
1359                            ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
1360                            ;*  :  THEN-REPORT OPERATOR ERROR
1361                            ;*  :  ELSE-PROCEED
1362                            ;*  ENDIF
1363                            ;*  BGNDO
1364                            ;*  :  CLEAR ALL PORTS
1365                            ;*  :  INIT THE SYSTEM
1366                            ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
1367                            ;*  :  INIT THE PARAMETER TABLE POINTER - PTABLE
1368                            ;*  :  SET THE BYTE COUNT & DATA PACKING FORMAT FROM 'PTABLE' DATA
1369                            ;*  :  INIT COUNTERS + CONTROLS
1370                            ;*  :  LOAD BYTE COUNTER WITH EXPECTED # BYTES FROM READ PATH
1371                            ;*  :  RESTART WRITE MICROCONTROLLER
1372                            ;*  :  CLOCK WMC THROUGH RESTART CODE
1373                            ;*  :  START 'GCR' READ DATA XFR
1374                            ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
1375                            ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1376                            ;*  :  CLOCK THE FIFO'S
1377                            ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
1378                            ;*  :  WAIT
1379                            ;*  :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1380                            ;*  :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
1381                            ;*      (END-MARK, RESIDUAL GRP, CRC GRP, MARK-2)
```

```
1382                                    ;*  :  SETUP A WATCHDOG TIMER
1383                                    ;*  :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
1384                                    ;*  :  IF WATCHDOG TIMER=0
1385                                    ;*  :  :  THEN-REPORT NO 'SCLK'
1386                                    ;*  :  :  ELSE-CONT
1387                                    ;*  :  ENDIF
1388                                    ;*  :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
1389                                    ;*  :  IF SCLK STILL UP AFTER 5000. CLOCKS
1390                                    ;*  :  :  THEN - CHECK BYTE COUNTER TO SEE IF -0
1391                                    ;*  :  :  :  IF BYTE COUNTER =0
1392                                    ;*  :  :  :  :  THEN - DROP WMC ENABLE
1393                                    ;*  :  :  :  :  ELSE - CONTINUE
1394                                    ;*  :  :  :  ENDIF
1395                                    ;*  :  :  ELSE - REPORT ERROR ... SCLK WON'T CLEAR AFTER SETTING
1396                                    ;*  :  ENDIF
1397                                    ;*  :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
1398                                    ;*  :  IF 18 BIT DATA SAME AS EXPECTED
1399                                    ;*  :  :  THEN-PROCEED
1400                                    ;*  :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
1401                                    ;*  :  ENDIF
1402                                    ;*  :  POINT TO NEXT ENTRY IN 'PTABLE' STORAGE
1403                                    ;*  :  IF ALL DONE WITH DATA TRANSFERS IN 'PTABLE'
1404                                    ;*  :  :  THEN - DECREMENT THE ITERATION COUNTER
1405                                    ;*  :  :     IF ALL ITERATIONS FINISHED
1406                                    ;*  :  :     :  THEN-EXIT
1407                                    ;*  :  :     :  ELSE-CONTINUE
1408                                    ;*  :  :     ENDIF
1409                                    ;*  :  :  ELSE-CONTINUE
1410                                    ;*  :  ENDIF
1411                                    ;*  :  DO UNTIL ALL ITERATIONS EXECUTED
1412                                    ;*  ENDDO
1413                                    ;*ENDTST
1414  4300                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1415                                    ;*WMC7 MICRO TEST 01
1416                                    ;*WMC7 MICRO ERROR 01
1417                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1418                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1419                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1420                                    ;*FATAL ERROR - TEST ABORTED
1421                                    ;*
1422                                    ;*WMC7 MICRO TEST 01
1423                                    ;*WMC7 MICRO ERROR 02
1424                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1425                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1426                                    ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET ... DATA NOT READY FROM BYTE ASSY
1427                                    ;*DATA HAS BEEN SENT TO WMC FROM CRC BOARD - 'XFER' SEEN IN WMCSTA WORD
1428                                    ;*
1429                                    ;*WMC7 MICRO TEST 01
1430                                    ;*WMC7 MICRO ERROR 03
1431                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1432                                    ;*M8959 (WMC) & ALL OTHER BOARDS
```

```
                                        ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA WORD ... DATA NOT READY
                                        ;*AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78 - NO 'XFER' SEEN IN
                                        ;*WMCSTA WORD. DATA NOT SEEN FROM CRC LOGIC.
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 04
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
                                        ;*DURING A READ FORWARD COMMAND
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 05
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 06
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
                                        ;*ACTUAL   NNNN
                                        ;*EXPECTED   NNNN
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 07
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
                                        ;*ACTUAL - NNNN
                                        ;*EXPECTED - NNNN
4300                                    S
 (1)                                    ; ************************************************************************************
                                        TEST1:  TESTX   @1
 (1) 4300  3E  01          7.0                         MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
                                        ;%WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;&M8959 (WMC) & ALL OTHER BOARDS
                                        ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING

                                                REQ     7
 (1) 4305  CD  06  28     18.0                         CALL    REQST
 (1) 4308  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (1) 4309  00  00                                      .WORD                   ;SYSTEM '*' COUNT
 (1) 430B  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (1) 430E  07                                          .BYTE   7               ;REQUEST CODE
430F                                            RIN     R12L                    ;GET THE SELECTED PORT FROM 'HOST'
 (1) 430F  DB  94         10.0                         IN      R12L                    ;READ R12L INTO AC
 (1) 4311  7F              4.0                         MOV     A,A                     ;RETRY LINK
```

```
1476  4312  32  03  48   13.0          STA     UNITMP          ;SAVE IT
1477  4315  A7               4.0       ANA     A               ;SET THE CONDITION BITS
1478  4316  C2  27  43      10.0       JNZ     FOUND           ;SEE IF ANY WERE SELECTED IN 'HOST'
1479  4319                             ERR     EXIT,NOUNIT
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                               MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                 .BYTE
 (1)  431E  CD  15  28      18.0       NOJNIT::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  F8  45      10.0                 JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1480                        ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
1481                        ;>FATAL ERROR - TEST ABORTED
1482  4324  C3  F8  45      10.0         JMP     EXIT
1483                        ;HERE IF THE HOST CPU INDICATED A PORT TO USE
1484
1485  4327  CD  4F  47      18.0  FOUND: CALL    CLEAR           ;INIT ALL THE PORTS IN THE PORT BOARD
1486  432A  3A  03  48      13.0         LDA     UNITMP          ;GET THE RESPONSE FROM 'HOST' CPU
1487  432D  06  00           7.0         MVI     B,0             ;INIT 'B'
1488  432F  E6  01           7.0         ANI     1               ;IS UNIT 0 SELECTED?
1489  4331  C2  47  43      10.0         JNZ     FPORT           ;YUP - FOUND THE PORT TO USE
1490  4334  04               4.0         INR     B               ;NO - COUNT THE PORT
1491  4335  3A  03  48      13.0         LDA     UNITMP          ;GET RESPONSE AGAIN
1492  4338  E6  02           7.0         ANI     2               ;PORT 1 SELECTED?
1493  433A  C2  47  43      10.0         JNZ     FPORT           ;YUP
1494  433D  04               4.0         INR     B               ;COUNT THE PORT
1495  433E  3A  03  48      13.0         LDA     UNITMP          ;GET RESPONSE
1496  4341  E6  04           7.0         ANI     4               ;PORT 2 SELECTED?
1497  4343  C2  47  43      10.0         JNZ     FPORT           ;YUP
1498  4346  04               4.0         INR     B               ;NO - ASSUME PORT 3
1499
1500  4347  78               4.0  FPORT: MOV     A,B             ;PUT UNIT # IN AC
1501  4348  32  04  48      13.0         STA     SUNIT           ;SAVE THE UNIT UNDER TEST
1502  434B  DB  E0           10.0         IN      INTSTA          ;GET THE PORT SELECT STATUS
1503  434D  E6  80           7.0         ANI     BIT7
1504  434F  B0               4.0         ORA     B               ;ADD IN THE SELECTED PORT #
1505  4350  D3  E0           10.0         OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
1506  4352  3E  60           7.0         MVI     A,P.LWR!P.LCS   ;GET THE LOOP-COMMAND-STATUS BIT
1507  4354  D3  48           10.0         OUT     PDIAG           ;SELECT IT
1508  4356  AF               4.0         XRA     A
1509  4357  D3  44           10.0         OUT     TAMT            ;CLEAR THE AMTIE LINES
1510  4359  D3  D2           10.0         OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS
1511  435B  D3  D2           10.0         OUT     TRKENA
1512  435D  3E  04           7.0         MVI     A,P.RPEN        ;SET READ PATH ENABLE
1513  435F  D3  4C           10.0         OUT     PENAB           ;NOW
1514  4361  3E  10           7.0         MVI     A,RDCLK         ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1515  4363  D3  F0           10.0         OUT     CLKCTL
1516  4365  D3  C0           10.0         OUT     DBUSCTL         ;CLEAR THE DATA BUS CONTROL WORD
1517  4367  21  4E  46      10.0         LXI     H,XFRPTB        ;POINT TO THE 1ST DATA XFR PARAMETER TABLE
1518  436A  22  08  48      16.0         SHLD    PTABLE          ;SAVE THE TABLE ADDRESS
1519                        ;INIT THE COUNTER CONTROL AND COUNTERS
1520
1521  436D                  T1LOOP: INICNT
 (1)  436D  3E  30           7.0                         MVI     A,@060          ;INIT THE
```

```
(1)  436F  D3  D7          10.0                    OUT   CNTCTL          ;BYTE COUNTER
(1)  4371  3E  70           7.0                    MVI   A,@160          ;INIT THE
(1)  4373  D3  D7          10.0                    OUT   CNTCTL          ;PAD COUNTER
(1)  4375  3E  B0           7.0                    MVI   A,@260          ;INIT THE
(1)  4377  D3  D7          10.0                    OUT   CNTCTL          ;ERROR COUNTER
1522 4379                                  CLRECT
(1)  4379  AF               4.0                    XRA   A               ;CLEAR THE ACCUMULATOR
(1)  437A  D3  D6          10.0                    OUT   ERRCNT          ;CLEAR BITS 7-0
(1)  437C  D3  D6          10.0                    OUT   ERRCNT          ;CLEAR BITS 15-8
1523
1524 437E  2A  08  48      16.0            LHLD   PTABLE                 ;GET THE PARAMETER POINTER
1525 4381  7E               7.0            MOV    A,M                    ;GET THE LOW BYTE OF EXPECTED ADDRESS
1526 4382  32  01  48      13.0            STA    EXPADR                 ;SAVE IT
1527 4385  23               6.0            INX    H                      ;POINT TO THE HIGH BYTE OF ADDRESS
1528 4386  7E               7.0            MOV    A,M                    ;GET THE HIGH BYTE
1529 4387  32  02  48      13.0            STA    EXPADR+1               ;SAVE IT
1530 438A  23               6.0            INX    H                      ;POINT TO THE PACKING MODE NUMBER
1531 438B  7E               7.0            MOV    A,M                    ;GET IT
1532 438C  32  FF  47      13.0            STA    FMTNUM                 ;SAVE IT
1533 438F  23               6.0            INX    H                      ;POINT TO THE BYTE COUNT FOR THIS XFR
1534 4390  7E               7.0            MOV    A,M                    ;GET IT
1535 4391  32  00  48      13.0            STA    BCNT                   ;SAVE IT
1536 4394  23               6.0            INX    H                      ;POINT TO THE # SCLK'S TO CHECK IN XFR
1537 4395  7E               7.0            MOV    A,M                    ;GET THE COUNT
1538 4396  32  FE  47      13.0            STA    ESCLK                  ;SAVE THE NUMBER
1539 4399  23               6.0            INX    H                      ;POINT TO THE 1ST WORD IN NEXT ENTRY
1540 439A  22  0A  48      16.0            SHLD   NPTABLE                ;SAVE THE ADDRESS OF NEW TABLE
1541
1542 439D  3A  FF  47      13.0            LDA    FMTNUM                 ;GET THE PACKING MODE # AGAIN
1543 43A0  07               4.0            RLC
1544 43A1  07               4.0            RLC
1545 43A2  07               4.0            RLC
1546 43A3  07               4.0            RLC
1547 43A4  E6  70           7.0            ANI    $70                    ;POSITION THE FMT #
1548 43A6  D3  D0          10.0            OUT    DATACTL                ;LOAD THE DATA CONTROL WORD
1549 43A8                                  ROUT   R02H                   ;SAVE FOR ERROR PRINTING
(1)  43A8  D3  85          10.0                   OUT    R02H                    ;WRITE AC INTO R02H
(1)  43AA  7F               4.0                   MOV    A,A                     ;RETRY LINK
1550 43AB  3E  9B           7.0            MVI    A,@233                 ;SET THE DDR CONTROL TO 'IN'
1551 43AD  D3  DB          10.0            OUT    DDRCTL                 ;TO READ THE 'DDR'
1552 43AF  06  06           7.0            MVI    B,6                    ;SETUP TO CLEAR THE RESIDUAL GROUP
1553 43B1  AF               4.0            XRA    A                      ;DATA STORAGE FOR THIS XFR
1554 43B2  21  3E  46      10.0            LXI    H,RGRP                 ;POINT TO THE RESID GRP AREA
1555 43B5  77               7.0  T1RCLR:  MOV    M,A                    ;CLEAR A LOCATION IN THE RESID GRP AREA
1556 43B6  23               6.0            INX    H                      ;POINT TO THE NEXT LOCATION
1557 43B7  05               4.0            DCR    B                      ;DECREMENT THE LOOP COUNT
1558 43B8  C2  B5  43      10.0            JNZ    T1RCLR                 ;LOOP TILL ALL BYTES CLEARED
1559 43BB  21  3E  46      10.0            LXI    H,RGRP                 ;RESET THE POINTER
1560 43BE  3E  01           7.0            MVI    A,1                    ;GET THE 1ST DATA BYTE IN THE RESID GRP
1561 43C0  77               7.0  T1RFIL:  MOV    M,A                    ;INSERT THE DATA BYTE
1562 43C1  23               6.0            INX    H                      ;POINT TO THE NEXT LOCATION
1563 43C2  E5              12.0            PUSH   H                      ;SAVE H & L
1564 43C3  21  00  48      10.0            LXI    H,BCNT                 ;POINT TO THE BYTE COUNT STORAGE
1565 43C6  BE               7.0            CMP    M                      ;SEE IF DONE FILLING
```

```
1566  43C7  E1              10.0          POP   H
1567  43C8  CA  CF  43      10.0          JZ    T1FOK          ;JUMP IF RESID GRP BUILT
1568  43CB  3C               4.0          INR   A              ;UPDATE DATA BYTE FOR RESID GRP STORAGE
1569  43CC  C3  C0  43      10.0          JMP   T1RFIL         ;LOOP TILL DONE
1570
1571  43CF  3A  00  48      13.0  T1FOK:  LDA   BCNT           ;GET THE DESIRED BYTE COUNT FOR XFR
1572  43D2  32  23  4F      13.0          STA   BYTEL
1573  43D5  AF               4.0          XRA   A
1574  43D6  32  24  4F      13.0          STA   BYTEH
1575  43D9  CD  46  00      18.0          CALL  LKMOD7         ;CALC THE MOD7 REMAINDER
1576  43DC  3A  00  48      13.0          LDA   BCNT
1577  43DF  3D               4.0          DCR   A
1578  43E0  E6  1F           7.0          ANI   @37            ;THIS IS THE MOD32 NUMBER
1579  43E2  57               4.0          MOV   D,A            ;SAVE IT
1580  43E3  78               4.0          MOV   A,B            ;GET THE MOD7 NUMBER
1581  43E4  0F               4.0          RRC                  ;POSITION IT TO LOOK LIKE THE MOD7
1582  43E5  0F               4.0          RRC                  ;FOR THE CRC DATA GROUP
1583  43E6  0F               4.0          RRC
1584  43E7  B2               4.0          ORA   D              ;ADD IN THE MOD32 NUMBER
1585  43E8  32  4C  46      13.0          STA   RCHRWD         ;SAVE IT IN THE CRC DATA GROUP AREA
1586  43EB  3A  00  48      13.0          LDA   BCNT           ;GET THE BYTE COUNT AGAIN
1587  43EE  D3  D4          10.0          OUT   BYTCNT         ;LOAD BYTE COUNTER WITH EXPECTED # BYTES
1588  43F0  AF               4.0          XRA   A              ;FROM THE READ PATH LOGIC
1589  43F1  D3  D4          10.0          OUT   BYTCNT
1590  43F3  D3  D5          10.0          OUT   PADCNT         ;FAKE OUT THE PAD COUNTER
1591  43F5  D3  D5          10.0          OUT   PADCNT
1592  43F7                                ROUT  R02L           ;CLEAR OUT THE REST OF REG 2 IN CAS
 (1)  43F7  D3  84          10.0                OUT   R02L             ;WRITE AC INTO R02L
 (1)  43F9  7F               4.0                MOV   A,A              ;RETRY LINK
1593  43FA                                ROUT  R05L           ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
 (1)  43FA  D3  8A          10.0                OUT   R05L             ;WRITE AC INTO R05L
 (1)  43FC  7F               4.0                MOV   A,A              ;RETRY LINK
1594  43FD  32  FD  47      13.0          STA   SCLKCT         ;INIT THE SCLK DETECTED COUNTER
1595  4400  32  07  48      13.0          STA   ONCE           ;CLEAR THE BYTE COUNTER POSITIVE FLAG
1596                                ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1597
1598  4403  3E  01           7.0          MVI   A,W.RST        ;GET THE WMC RESTART BIT
1599  4405  D3  D3          10.0          OUT   WMCCTL         ;SET RESTART
1600  4407  3E  15           7.0          MVI   A,SSTEP!RDCLK  ;GET SINGLE STEP FOR WMC
1601  4409  D3  F0          10.0          OUT   CLKCTL         ;SET SINGLE STEP MODE
1602  440B  32  05  48      13.0          STA   CCTLWD         ;SAVE THE WORD
1603  440E  CD  5B  48      18.0          CALL  CLKSYS         ;CLOCK THE SYSTEM TO CAUSE RESTART TO
1604  4411  CD  5B  48      18.0          CALL  CLKSYS         ;TO TAKE HOLD
1605  4414  AF               4.0          XRA   A
1606  4415  D3  D3          10.0          OUT   WMCCTL         ;FINISH THE RESTART CYCLE
1607
1608                                ;CLOCK THE WMC THRU ITS RESTART AREA
1609
1610  4417  06  B4           7.0          MVI   B,180          ;GET A LOOP COUNT
1611  4419  CD  5B  48      18.0  CLKRS:  CALL  CLKSYS         ;CLOCK THE SYSTEM
1612  441C  05               4.0          DCR   B              ;DECREMENT THE LOOP COUNTER
1613  441D  C2  19  44      10.0          JNZ   CLKRS          ;CLOCK TILL COUNT RUN OUT
1614  4420  3E  90           7.0          MVI   A,W.GCR.W.ENAB ;SETUP FOR A GCR DATA XFR
1615  4422  D3  D3          10.0  T1ST:   OUT   WMCCTL         ;LOAD THE WORD IN WMC
```

```
1616
1617                                        ;CALCULATE THE RESIDUAL DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1618
1619  4424  21  3E  46        10.0  C1EC1:  LXI   H,RGRP          ;POINT THE THE 1ST DATA GROUP DATA TABLE
1620  4427  06  07              7.0         MVI   B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1621  4429  CD  1C  48        18.0         CALL  CLECC           ;INIT THE ECC CHARACTER
1622
1623  442C  7E                  7.0  C1EC1L: MOV   A,M             ;GET A DATA BYTE FROM THE TABLE
1624  442D  CD  21  48        18.0         CALL  ECC             ;CALC ECC WITH THIS CHARACTER
1625  4430  23                  6.0         INX   H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1626  4431  05                  4.0         DCR   B               ;DECREMENT THE COUNTER OF DATA BYTES
1627  4432  C2  2C  44        10.0         JNZ   C1EC1L          ;STAY HERE TILL ALL DATA BYTES USED
1628  4435  3A  48  48        13.0         LDA   ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1629  4438  32  45  46        13.0         STA   RGECC           ;SAVE THE CALCULATED ECC CHARACTER
1630
1631                                        ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1632
1633  443B  21  46  46        10.0  C1EC2:  LXI   H,CRCGRP        ;POINT THE THE 2ND DATA GROUP DATA TABLE
1634  443E  06  07              7.0         MVI   B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1635  4440  CD  1C  48        18.0         CALL  CLECC           ;INIT THE ECC CHARACTER
1636
1637  4443  7E                  7.0  C1EC2L: MOV   A,M             ;GET A DATA BYTE FROM THE TABLE
1638  4444  CD  21  48        18.0         CALL  ECC             ;CALC ECC WITH THIS CHARACTER
1639  4447  23                  6.0         INX   H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1640  4448  05                  4.0         DCR   B               ;DECREMENT THE COUNTER OF DATA BYTES
1641  4449  C2  43  44        10.0         JNZ   C1EC2L          ;STAY HERE TILL ALL DATA BYTES USED
1642  444C  3A  48  48        13.0         LDA   ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1643  444F  32  4D  46        13.0         STA   CRCECC          ;SAVE THE CALCULATED ECC CHARACTER
1644  4452  3E  A8              7.0  T1PLO:  MVI   A,R.PLO1!R.TBJN!R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1645                                                              ;SET TIE BUS JAM AND PLO DISABLE
1646  4454  D3  09            10.0         OUT   RPCTL
1647  4456  AF                  4.0         XRA   A
1648  4457  D3  0A            10.0         OUT   RTIEB           ;CLEAR THE TIE BUS
1649  4459  D3  08            10.0         OUT   RFIFOL          ;CLOCK THE FIFO'S
1650  445B  3E  0D              7.0         MVI   A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1651  445D  D3  0B            10.0         OUT   RCMD
1652
1653  445F  00                  4.0         NOP                   ;WAIT
1654  4460  00                  4.0         NOP                   ;WAIT
1655  4461  00                  4.0         NOP
1656  4462  00                  4.0         NOP
1657  4463  00                  4.0         NOP
1658
1659  4464  3E  A9              7.0         MVI   A,R.PLO1!R.STPC!R.PLOD.R.TBJN   ;STOP THE READ PATH
1660                                                              ;SET TIE BUS JAM AND PLO DISABLE
1661  4466  D3  09            10.0         OUT   RPCTL
1662                                  ;LOAD THE 'END MARK' INTO THE FIFO
1663
1664  4468  21  10  48        10.0  T1EMK:  LXI   H,TRNOUT         ;POINT TO THE 4X5 STORAGE AREA
1665  446B  06  05              7.0         MVI   B,5             ;SET A LOOP COUNT
1666  446D  3E  FF              7.0  T1EMKL: MVI   A,$FF           ;GET 8 TRKS OF DATA - ALL 1'S
1667  446F  77                  7.0         MOV   M,A             ;SAVE IN THE TABLE
1668  4470  23                  6.0         INX   H               ;POINT TO THE PARITY TRK AREA
1669  4471  3E  01              7.0         MVI   A,1
```

```
1670  4473  77              7.0          MOV     M,A             ;SET IT TO 1 ALSO
1671  4474  23              6.0          INX     H               ;POINT TO NEXT STORAGE AREA
1672  4475  05              4.0          DCR     B               ;DECREMENT THE LOOP COUNT
1673  4476  C2  6D  44     10.0          JNZ     T1EMKL          ;LOOP TILL END MARK IN STORAGE
1674  4479  CD  27  46     18.0          CALL    LOADIT          ;INSERT IT INTO THE FIFO
1675
1676  447C  21  3E  46     10.0  T1SG1:  LXI     H,RGRP          ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
1677  447F  CD  12  46     18.0          CALL    CONVRT          ;CONVERT SUBGROUP - 4X5
1678  4482  CD  27  46     18.0          CALL    LOADIT          ;INJECT THE DATA INTO THE FIFO
1679  4485  21  42  46     10.0  T1SG2:  LXI     H,RGRP2         ;POINT TO THE 2ND SUBGROUP OF DATA
1680  4488  CD  12  46     18.0          CALL    CONVRT          ;MAKE IT 4X5
1681  448B  CD  27  46     18.0          CALL    LOADIT          ;INJECT IT INTO FIFO
1682
1683  448E  21  46  46     10.0  T1SG3:  LXI     H,CRCGRP        ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
1684  4491  CD  12  46     18.0          CALL    CONVRT          ;MAKE IT 4X5
1685  4494  CD  27  46     18.0          CALL    LOADIT          ;INJECT THE DATA INTO FIFO
1686  4497  21  4A  46     10.0  T1SG4:  LXI     H,CRCG2         ;POINT TO THE 4TH SUBGROUP OF DATA
1687  449A  CD  12  46     18.0          CALL    CONVRT          ;MAKE IT 4X5
1688  449D  CD  27  46     18.0          CALL    LOADIT          ;INJECT THE DATA INTO FIFO
1689                                  ;LOAD THE MARK 2 SIGNAL INTO THE FIFO
1690  44A0  06  03           7.0  T1MK2:  MVI     B,3
1691  44A2  21  10  48      10.0          LXI     H,TRNOUT        ;POINT TO THE LOAD BUFFER
1692  44A5  3E  FF           7.0  T1MK2L: MVI     A,$FF
1693  44A7  77               7.0          MOV     M,A             ;SAVE 8 TRKS
1694  44A8  23               6.0          INX     H
1695  44A9  3E  01           7.0          MVI     A,1
1696  44AB  77               7.0          MOV     M,A             ;SAVE THE PARITY TRK
1697  44AC  23               6.0          INX     H
1698  44AD  05               4.0          DCR     B               ;DECREMENT THE LOOP COUNTER
1699  44AE  C2  A5  44      10.0          JNZ     T1MK2L          ;LOOP TILL DONE
1700  44B1  06  04           7.0          MVI     B,4
1701  44B3  AF               4.0          XRA     A
1702  44B4  77               7.0  T1M2L2: MOV     M,A             ;SAVE 8 TRKS OF 0
1703  44B5  23               6.0          INX     H
1704  44B6  05               4.0          DCR     B               ;DECREMENT LOOP COUNTER
1705  44B7  C2  B4  44      10.0          JNZ     T1M2L2          ;LOOP TILL ALL MARK - 2 IN BUFFER
1706  44BA  CD  27  46      18.0          CALL    LOADIT          ;STUFF MARK - 2 IN THE FIFO
1707
1708                                  ;HERE TO START THE DIAGNOSTIC READ COMMAND IN THE READ PATH TO
1709                                  ;CAUSE THE DATA IN THE FIFO TO GO THRU READ PATH & ENTER THE
1710                                  ;WMC BYTE ASSY LOGIC VIA THE CRC BOARD
1711
1712  44BD  3E  0B           7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
1713  44BF  D3  08          10.0          OUT     RCMD            ;COMMAND
1714
1715  44C1  11  01  00      10.0  T1CLKA: LXI     D,1             ;SET WATCH DOG INCREMENT
1716  44C4  21  68  C5      10.0          LXI     H,-15000        ;SET WATCH DOG COUNT TO 15000
1717  44C7  AF               4.0          XRA     A
1718  44C8  32  06  48      13.0          STA     HAVEXFR         ;CLEAR THE HAVE 'XFER' BIT FLAG
1719  44CB  D3  0C          10.0  T1CLK:  OUT     RINST           ;STEP THE READ PATH
1720  44CD  CD  5E  48      18.0          CALL    CLKSYS          ;CLOCK THE WMC
1721  44D0  DB  D1          10.0          IN      TUSEL0          ;SEE IF BYTE COUNTER WENT TO 0
1722  44D2  E6  40           7.0          ANI     BIT6            ;SO THE WMC CAN BE DISABLED TO FINISH XFR
1723  44D4  C2  DB  44      10.0          JNZ     T1CLK1          ;JUMP IF BYTE TERMINAL COUNTER STILL - 0
```

```
1724  44D7  3E  10        7.0          MVI   A,W.GCR          ;DROP THE WMC ENABLE BIT
1725  44D9  D3  D3       10.0          OUT   WMCCTL           ;IN THE WMC CONTROL WORD
1726
1727  44DB  DB  D0       10.0  T1CLK1: IN    WMCSTA           ;SEE IF EVER GET 'XFER' BIT
1728  44DD  E6  20        7.0          ANI   BIT5             ;SAVE JUST 'XFER' BIT IN STATUS
1729  44DF  CA  E5  44   10.0          JZ    T1CLK2           ;JUMP IF NOT UP
1730  44E2  32  06  48   13.0          STA   HAVEXFR          ;SET THE 'XFER' FLAG
1731
1732  44E5  DB  C0       10.0  T1CLK2: IN    DBUSSTA          ;GET THE DATABUS STATUS
1733  44E7  E6  02        7.0          ANI   T.SCLK           ;SEE IF SCLK UP - DATA READY
1734  44E9  C2  0D  45   10.0          JNZ   T1SCLK           ;JUMP IF SCLK UP
1735
1736  44EC  19           10.0  T1CLK3: DAD   D
1737  44ED  D2  CB  44   10.0          JNC   T1CLK            ;STAY IN LOOP UNLESS TIMEOUT
1738
1739  44F0  3A  06  48   13.0          LDA   HAVEXFR          ;SEE IF EVER GOT 'XFER' BIT SET
1740  44F3  A7            4.0          ANA   A
1741  44F4  C2  FC  44   10.0          JNZ   DUMM1A           ;JUMP IF NO SCLK EVER BUT HAVE 'XFER'
1742  44F7                             ERR   T1LOOP,DUMM1A,3
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F7  CD  09  28   18.0                    CALL  ERLP           ;PROCESS ERROR - DO 2.3
 (1)       0002                                 MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44FA  02                                   .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  44FB  03                                   .BYTE 3
 (1)  44FC  CD  15  28   18.0          DUMM1A::  CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FF  DA  6D  43   10.0                    JC    T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1743                                 ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET ... DATA NOT READY FROM BYTE ASSY
1744                                 ;>DATA HAS BEEN SENT TO WMC FROM CRC BOARD - 'XFER' SEEN IN WMCSTA WORD
1745  4502                           T1ERNS: ERR   T1LOOP,DUMM1,3
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4502  CD  09  28   18.0                    CALL  ERLP           ;PROCESS ERROR - DO 2.3
 (1)       0003                                 MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4505  03                                   .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  4506  03                                   .BYTE 3
 (1)  4507  CD  15  28   18.0          DUMM1::   CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  450A  DA  6D  43   10.0                    JC    T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1746                                 ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA WORD ... DATA NOT READY
1747                                 ;>AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78 - NO 'XFER' SEEN IN
1748                                 ;>WMCSTA WORD. DATA NOT SEEN FROM CRC LOGIC.
1749
1750                                 ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO =0
1751                                 ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
1752                                 ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDI ATING THE DATA IS
1753                                 ;ON THE MASSBUS
1754
1755  450D  11  01  00   10.0  T1SCLK: LXI   D,1
1756  4510  21  78  EC   10.0          LXI   H,-5000          ;SETUP THE DROP DEAD TIMER
1757
1758  4513  D3  0C       10.0  T1SCKL: OUT   RINST            ;CLOCK THE READ PATH ONCE
1759  4515  CD  5B  48   18.0          CALL  CLKSYS           ;CLOCK THE WMC TO CLEAR 'SCLK'
1760  4518  DB  C0       10.0          IN    DBUSSTA          ;SEE IF 'SCLK' IS GONE
1761  451A  E6  02        7.0          ANI   T.SCLK           ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
1762  451C  CA  43  45   10.0          JZ    DUMM2            ;JUMP IF 'SCLK' - 0
1763  451F  19           10.0          DAD   D                ;DECREMENT THE TIMER
```

```
1764  4520  D2  13  45      10.0          JNC     T1SCKL          ;LOOP AGAIN IF STILL TIME LEFT
1765
1766  4523  DB  D4         10.0          IN      BYTCNT          ;GET LOW BITS OF BYTE COUNTER
1767  4525  47              4.0          MOV     B,A             ;SAVE IT
1768  4526  DB  D4         10.0          IN      BYTCNT          ;THEN HIGH BITS
1769  4528  78              4.0          MOV     A,B             ;GET THE LOW BITS AGAIN
1770  4529  A7              4.0          ANA     A
1771  452A  C2  3E  45      10.0          JNZ     T1SCKE          ;JUMP IF POSITIVE - ERROR CONDITION
1772  452D  3A  07  48      13.0          LDA     ONCE            ;SEE IF HERE BEFORE
1773  4530  A7              4.0          ANA     A               ;FOR THIS SAME DATA XFR
1774  4531  C2  3E  45      10.0          JNZ     T1SCKE          ;WHOOPS...ERROR DETECTED.
1775  4534  3E  10          7.0          MVI     A,W.GCR         ;CLEAR THE WMC ENABLE BIT SO WMC WILL
1776  4536  D3  D3         10.0          OUT     WMCCTL          ;FINISH THE DATA XFR
1777  4538  32  07  48     13.0          STA     ONCE            ;SET THE ONCE FLAG
1778  4538  C3  0D  45      10.0          JMP     T1SCLK          ;BACK TO CLOCK LOOP
1779
1780  453E                          T1SCKE: ERR     T1LOOP,DUMM2,3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453E  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                          MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4541  04                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4542  03                            .BYTE   3
 (1)  4543  CD  15  28      18.0          DUMM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4546  DA  6D  43      10.0                  JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1781                                ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1782                                ;>DURING A READ FORWARD COMMAND
1783                                ;HERE WHEN 'SCLK' WAS SET - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
1784                                ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
1785
1786  4549  DB  D8         10.0  T1SCK0: IN      DDRA            ;GET THE DDR A DATA
1787  454B  32  FA  47      13.0          STA     SDDRA           ;SAVE IT
1788  454E                            ROUT    R16L            ;SAVE FOR PRINTING
 (1)  454E  D3  9C         10.0                  OUT     R16L            ;WRITE AC INTO R16L
 (1)  4550  7F              4.0                  MOV     A,A             ;RETRY LINK
1789  4551  DB  D9         10.0          IN      DDRB            ;GET DDR B DATA
1790  4553  32  FB  47      13.0          STA     SDDRB           ;SAVE IT
1791  4556                            ROUT    R16H            ;SAVF FOR ERROR REPORTING
 (1)  4556  D3  9D         10.0                  OUT     R16H            ;WRITE AC INTO R16H
 (1)  4558  7F              4.0                  MOV     A,A             ;RETRY LINK
1792  4559  DB  DA         10.0          IN      DDRC            ;GET DDR C DATA
1793  455B  E6  03          7.0          ANI     3
1794  455D  32  FC  47      13.0          STA     SDDRC           ;SAVE IT
1795  4560  F6  80          7.0          ORI     BIT7            ;ADD IN THE PRINT FLAG
1796  4562                            ROUT    R17L            ;SAVE IT FOR PRINTING
 (1)  4562  D3  9E         10.0                  OUT     R17L            ;WRITE AC INTO R17L
 (1)  4564  7F              4.0                  MOV     A,A             ;RETRY LINK
1797
1798  4565  3A  FD  47      13.0          LDA     SCLKCT          ;GET THE CURRENT SCLK COUNT
1799  4568  3C              4.0          INR     A               ;COUNT THIS SCLK
1800  4569  32  FD  47      13.0          STA     SCLKCT          ;SAVE THE NEW COUNT
1801  456C                            ROUT    R05L            ;ALSO IN CAS FOR ERROR REPORTING
 (1)  456C  D3  8A         10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  456E  7F              4.0                  MOV     A,A             ;RETRY LINK
1802  456F  AF              4.0          XRA     A
```

```
1803  4570                                          ROUT    R05H
 (1)  4570  D3  8B          10.0                    OUT     R05H          ;WRITE AC INTO R05H
 (1)  4572  7F               4.0                    MOV     A,A           ;RETRY LINK
1804
1805                                    ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED
1806
1807  4573  2A  01  48       16.0  T1CKA:  LHLD    EXPADR                ;POINT TO THE TABLE OF EXPECTED DATA
1808  4576  7E                7.0          MOV     A,M                   ;GET BITS 7-0
1809  4577                               ROUT    R14L                    ;SAVE FOR ERROR REPORTING
 (1)  4577  D3  98          10.0                    OUT     R14L          ;WRITE AC INTO R14L
 (1)  4579  7F               4.0                    MOV     A,A           ;RETRY LINK
1810  457A  23                6.0          INX     H                     ;POINT TO NEXT BYTE
1811  457B  7E                7.0          MOV     A,M                   ;GET BITS 15-8
1812  457C                               ROUT    R14H                    ;SAVE IT
 (1)  457C  D3  99          10.0                    OUT     R14H          ;WRITE AC INTO R14H
 (1)  457E  7F               4.0                    MOV     A,A           ;RETRY LINK
1813  457F  23                6.0          INX     H
1814  4580  7E                7.0          MOV     A,M                   ;GET BITS 17-16
1815  4581  F6  80             7.0          ORI     BIT7                  ;ADD IN THE PRINT FLAG
1816  4583                               ROUT    R15L                    ;SAVE FOR ERROR REPORT
 (1)  4583  D3  9A          10.0                    OUT     R15L          ;WRITE AC INTO R15L
 (1)  4585  7F               4.0                    MOV     A,A           ;RETRY LINK
1817  4586  3A  FA  47       13.0          LDA     SDDRA                 ;GET THE DDR A DATA
1818  4589                               ROUT    ADATA                   ;SAVE IN CASE OF ERROR DETECTED
 (1)  4589  D3  94          10.0                    OUT     ADATA         ;WRITE AC INTO ADATA
 (1)  458B  7F               4.0                    MOV     A,A           ;RETRY LINK
1819  458C  2A  01  48       16.0          LHLD    EXPADR                ;POINT TO THE EXPECTED DATA TABLE
1820  458F  BE                7.0          CMP     M                     ;ACTUAL DATA = EXPECTED?
1821  4590  CA  9C  45       10.0          JZ      T1CKAC                ;JUMP IF OK
1822  4593  7E                7.0          MOV     A,M                   ;GET THE EXPECTED DATA
1823  4594                               ROUT    EDATA                   ;SAVE FOR ERROR REPORT
 (1)  4594  D3  95          10.0                    OUT     EDATA         ;WRITE AC INTO EDATA
 (1)  4596  7F               4.0                    MOV     A,A           ;RETRY LINK
1824  4597                               ERRB    T1LOOP,T1CKAC,@10
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4597  CD  12  28       18.0                    CALL    ERLPB         ;PROCESS ERROR - DO 2.3
 (1)        0005                                    MSGN    =       MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  459A  05                                      .BYTE   MSGN          ;MESSAGE NUMBER ID
 (1)  459B  08                                      .BYTE   @10           ;PRINT ROUTINE NUMBER
 (1)  459C  CD  15  28       18.0          T1CKAC::        CALL    CKLOP         ;CHECK LOOP FUNCTION - DO 2.2
 (1)  459F  DA  6D  43       10.0                    JC      T1LOOP        ;LOOP ADDRESS IF LOOP SPECIFIED
1825                                ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
1826  45A2  23                6.0  T1CKB:  INX     H                     ;POINT TO THE EXPECTED DATA FOR DDR B
1827  45A3  3A  FB  47       13.0          LDA     SDDRB                 ;GET THE DDR B DATA
1828  45A6                               ROUT    ADATA                   ;SAVE FOR ERROR REPORT
 (1)  45A6  D3  94          10.0                    OUT     ADATA         ;WRITE AC INTO ADATA
 (1)  45A8  7F               4.0                    MOV     A,A           ;RETRY LINK
1829  45A9  BE                7.0          CMP     M                     ;ACTUAL = EXPECTED DATA?
1830  45AA  CA  B6  45       10.0          JZ      T1CKBC                ;JUMP IF OK
1831  45AD  7E                7.0          MOV     A,M                   ;GET THE EXPECTED DATA
1832  45AE                               ROUT    EDATA                   ;SAVE FOR ERROR REPORT
 (1)  45AE  D3  95          10.0                    OUT     EDATA         ;WRITE AC INTO EDATA
 (1)  45B0  7F               4.0                    MOV     A,A           ;RETRY LINK
1833  45B1                               ERRB    T1LOOP,T1CKBC,@10
```

```
(1)
(1)  45B1  CD  12  28  18.0                ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
(1)                                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
(1)        0006                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1)  45B4  06                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
(1)  45B5  08                                  .BYTE   @10             ;PRINT ROUTINE NUMBER
(1)  45B6  CD  15  28  18.0                T1CKBC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1)  45B9  DA  6D  43  10.0                            JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1834                                       ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
1835
1836  45BC  23            6.0  T1CKC:  INX     H               ;POINT TO EXPECTED DDR C DATA
1837  45BD  3A  FC  47   13.0          LDA     SDDRC           ;GET DDR C DATA
1838  45C0                             ROUT    ADATA           ;SAVE FOR ERROR REPORT
(1)   45C0  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
(1)   45C2  7F             4.0                  MOV     A,A             ;RETRY LINK
1839  45C3  BE            7.0          CMP     M               ;ACTUAL DATA = EXPECTED DDRC DATA?
1840  45C4  CA  D0  45   10.0          JZ      T1CKCC          ;JUMP IF OK
1841  45C7  7E            7.0          MOV     A,M             ;GET THE EXPECTED DATA
1842  45C8                             ROUT    EDATA           ;SAVE FOR ERROR REPORT
(1)   45C8  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
(1)   45CA  7F             4.0                  MOV     A,A             ;RETRY LINK
1843  45CB                             ERRB    T1LOOP,T1CKCC,@10
(1)                                        ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
(1)   45CB  CD  12  28  18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
(1)        0007                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1)  45CE  07                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
(1)  45CF  08                                  .BYTE   @10             ;PRINT ROUTINE NUMBER
(1)  45D0  CD  15  28  18.0                T1CKCC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1)  45D3  DA  6D  43  10.0                            JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1844                                       ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
1845
1846  45D6  2A  01  48   16.0          LHLD    EXPADR          ;GET THE CURRENT EXPECTED DATA ADDR.
1847  45D9  23            6.0          INX     H
1848  45DA  23            6.0          INX     H               ;UPDATE THE EXPECTED ADDRESS POINTER
1849  45DB  23            6.0          INX     H               ;TO THE DATA
1850  45DC  22  01  48   16.0          SHLD    EXPADR          ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
1851  45DF  3A  FD  47   13.0          LDA     SCLKCT          ;GET THE SCLK COUNT
1852  45E2  21  FE  47   10.0          LXI     H,ESCLK         ;POINT TO THE EXPECTED SCLK COUNT
1853  45E5  BE            7.0          CMP     M               ;DONE WITH THIS TEST PASS?
1854  45E6  C2  C1  44   10.0          JNZ     T1CLKA          ;JUMP IF STILL EXPECTING MORE SCLKS
1855
1856                                     ;HERE TO SEE IF ALL XFRS IN THE 'PTABLE' HAVE BEEN DONE
1857
1858  45E9  2A  0A  48   16.0          LHLD    NPTABLE         ;POINT TO THE TABLE
1859  45EC  7E            7.0          MOV     A,M             ;GET THE 1ST WORD OF AN ENTRY
1860  45ED  FE  FF        7.0          CPI     @377            ;END OF THE TABLE?
1861  45EF  CA  F8  45   10.0          JZ      EXIT            ;EXIT IF END OF TABLE
1862  45F2  22  08  48   16.0          SHLD    PTABLE          ;SAVE ADDRESS OF NEXT XFR TABLE
1863  45F5  C3  6D  43   10.0          JMP     T1LOOP          ;EXECUTE THE NEXT XFR
1864
1865  45F8                       EXIT:   ENDTST  FOUND
(1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
(2)   45F8                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
(2)   45F8  CD  06  28  18.0                                   CALL    REQST
(2)   45FB  00                                                 .BYTE                   ;DATA PATTERN NUMBER
```

```
 (2)  45FC  00  00                                        .WORD                    ;SYSTEM '*' COUNT
 (2)  45FE  00  00                                        .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4600  00                                            .BYTE                    ;DATA COMPARE FLAG IF -1
 (2)  4601  07                                            .BYTE   7                ;REQUEST CODE
 (1)  4602  3A  9A  4F        13.0          LDA     ITERA                         ;GET ITERATION COUNT
 (1)  4605  3D                 4.0          DCR     A                             ;DOWNCOUNT
 (1)  4606  32  9A  4F        13.0          STA     ITERA                         ;SAVE COUNT
 (1)  4609  F2  27  43        10.0          JP      FOUND                         ;DO TEST UNTIL TILL - 0
1866  460C  CD  4F  47        18.0          CALL    CLEAR                         ;CLEAR ALL TU PORTS
1867  460F  C3  18  28        10.0          JMP     TSTEND
1868                                 ;HERE IS THE SUBROUTINES NECESSARY FOR CONVERTING SUBGROUPS AND LOADING
1869                                 ;THEM INTO THE FIFO
1870
1871  4612                           S
 (1)                               ; ***************************************************************************
1872                               ;*CONVRT -- THIS SUBROUTINE WILL GET A SUBGROUP OF DATA AS POINTED TO BY
1873                               ;*      THE H & L REGISTERS AND CAUSE THEM TO BE TRANSLATED (4X5). THE
1874                               ;*      TRANSLATED DATA WILL BE IN A TABLE CALLED 'TRNOUT'.
1875  4612                           S
 (1)                               ; ***************************************************************************
1876
1877  4612  D5                     12.0  CONVRT: PUSH    D                       ;SAVE D & E
1878  4613  11  0C  48             10.0          LXI     D,TRNIN                 ;POINT TO THE TEMP STORAGE FOR DATA BYTES
1879  4616  06  04                  7.0          MVI     B,4                     ;SETUP A LOOP COUNTER FOR 4 BYTES
1880
1881  4618  7E                      7.0  CNV1:   MOV     A,M                     ;GET A BYTE OF DATA
1882  4619  EB                      4.0          XCHG                            ;POINT TO THE STORAGE
1883  461A  77                      7.0          MOV     M,A                     ;SAVE DATA BYTE IN TEMP STORAGE
1884  461B  EB                      4.0          XCHG                            ;RESET POINTER
1885  461C  23                      6.0          INX     H                       ;POINT TO NEXT DATA BYTE
1886  461D  13                      6.0          INX     D                       ;POINT TO NEXT TEMP STORAGE ADDRESS
1887  461E  05                      4.0          DCR     B                       ;UPDATE THE LOOP COUNT
1888  461F  C2  18  46             10.0          JNZ     CNV1                    ;LOOP TILL 4 BYTES IN TEMP STORAGE
1889  4622  CD  7A  47             18.0          CALL    T4X5                    ;CONVERT THE 'SUBGROUP' OF DATA
1890  4625  D1                     10.0          POP     D
1891  4626  C9                     10.0          RET
1892  4627                           S
 (1)                               ; ***************************************************************************
1893                               ;*LOADIT -- THIS SUBROUTINE WILL TAKE THE TRANSLATED SUBGROUP (4X5)
1894                               ;*      DATA FROM TABLE 'TRNOUT' AND LOAD THE DATA INTO THE READ
1895                               ;*      PATH FIFO.
1896  4627                           S
 (1)                               ; ***************************************************************************
1897
1898  4627  21  10  48             10.0  LOADIT: LXI     H,TRNOUT                ;POINT TO THE TRANSLATED DATA TABLE
1899  462A  06  05                  7.0          MVI     B,5                     ;GET THE LOOP COUNT FOR SUBGROUP
1900
1901  462C  7E                      7.0  LDFIFO: MOV     A,M                     ;GET A BYTE OF DATA
1902  462D  D3  40                 10.0          OUT     TCMD                    ;STORE IT IN THE CMD REG FOR LOADING
1903  462F  23                      6.0          INX     H                       ;POINT TO THE PARITY BIT
1904  4630  7E                      7.0          MOV     A,M                     ;GET THE PARITY BIT
1905  4631  07                      4.0          RLC
1906  4632  F6  60                 7.0          ORI     P.LWR!P.LCS             ;ADD IN THE OTHER NECESSARY BITS
1907  4634  D3  48                10.0          OUT     PDIAG                   ;LOAD INTO CONTROL REG FOR LOADING
```

```
1908  4636  D3  08        10.0          OUT     RFIFOL          ;STROBE THE DATA INTO FIFO
1909  4638  23            6.0          INX     H               ;POINT TO THE NEXT SET OF DATA
1910  4639  05            4.0          DCR     B               ;DECREMENT THE LOOP COUNTER
1911  463A  C2  2C  46     10.0         JNZ     LDFIFO          ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
1912  463D  C9             10.0         RET
1913                                 ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
1914                                 ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ FORWARD'
1915
1916  463E  00                       RGRP:   .BYTE   0       ;RESIDUAL DATA GROUP DATA BYTE #1
1917  463F  00                               .BYTE   0
1918  4640  00                               .BYTE   0
1919  4641  00                               .BYTE   0
1920
1921  4642  00                       RGRP2:  .BYTE   0       ;2ND SUBGROUP IN RESIDUAL DATA GROUP
1922  4643  00                               .BYTE   0       ;LAST DATA BYTE IN RESIDUAL DATA GROUP
1923  4644  00                               .BYTE   0       ;THIS IS AN 'ACRC' DATA BYTE
1924  4645  00                       RGECC:  .BYTE   0       ;THIS IS THE ECC STORAGE FOR THE RESIDUAL DATA
1925
1926  4646  00                       CRCGRP: .BYTE   0       ;1ST BYTE IN CRC DATA GROUP
1927  4647  00                               .BYTE   0
1928  4648  00                               .BYTE   0
1929  4649  00                               .BYTE   0
1930
1931  464A  00                       CRCG2:  .BYTE   0       ;2ND SUBGROUP IN CRC DATA GROUP
1932  464B  00                               .BYTE   0
1933  464C  00                       RCHRWD: .BYTE   0       ;THIS IS THE RESIDUAL CHARACTER WORD
1934  464D  00                       CRCECC: .BYTE   0       ;ECC CHARACTER IN CRC SUBGROUP
1935                                 ;HERE IS THE PARAMETER TABLE FOR ALL DATA XFRS TO TEST THE SKIP COUNTS
1936                                 ;IN ALL DATA PACKING MODES
1937
1938  464E                         XFRPTB: XFRP    F11N,0,3,2
 (1)  464E  B9  46                        .ADDR   F11N    ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4650  00                            .BYTE   0       ;PACKING MODE 0
 (1)  4651  03                            .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  4652  02                            .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1939  4653                             XFRP    F15N,1,3,2
 (1)  4653  BF  46                        .ADDR   F15N    ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4655  01                            .BYTE   1       ;PACKING MODE 1
 (1)  4656  03                            .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  4657  02                            .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1940  4658                             XFRP    F10C1,2,1,2
 (1)  4658  C5  46                        .ADDR   F10C1   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  465A  02                            .BYTE   2       ;PACKING MODE 2
 (1)  465B  01                            .BYTE   1       ;BYTE COUNT THIS XFR
 (1)  465C  02                            .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1941  465D                             XFRP    F10C2,2,2,2
 (1)  465D  CB  46                        .ADDR   F10C2   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  465F  02                            .BYTE   2       ;PACKING MODE 2
 (1)  4660  02                            .BYTE   2       ;BYTE COUNT THIS XFR
 (1)  4661  02                            .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1942  4662                             XFRP    F10C3,2,3,2
 (1)  4662  D1  46                        .ADDR   F10C3   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4664  02                            .BYTE   2       ;PACKING MODE 2
 (1)  4665  03                            .BYTE   3       ;BYTE COUNT THIS XFR
```

```
  (1)  4666  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1943   4667                                  XFRP     F10D1,3,1,2
  (1)  4667  D7  46                                    .ADDR   F10D1   ;ADDRESS OF EXPECTED DATA TABLE
  (1)  4669  03                                        .BYTE   3       ;PACKING MODE 3
  (1)  466A  01                                        .BYTE   1       ;BYTE COUNT THIS XFR
  (1)  466B  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1944   466C                                  XFRP     F10D2,3,2,2
  (1)  466C  DD  46                                    .ADDR   F10D2   ;ADDRESS OF EXPECTED DATA TABLE
  (1)  466E  03                                        .BYTE   3       ;PACKING MODE 3
  (1)  466F  02                                        .BYTE   2       ;BYTE COUNT THIS XFR
  (1)  4670  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1945   4671                                  XFRP     F10D3,3,3,2
  (1)  4671  E3  46                                    .ADDR   F10D3   ;ADDRESS OF EXPECTED DATA TABLE
  (1)  4673  03                                        .BYTE   3       ;PACKING MODE 3
  (1)  4674  03                                        .BYTE   3       ;BYTE COUNT THIS XFR
  (1)  4675  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1946   4676                                  XFRP     F10D4,3,4,2
  (1)  4676  E9  46                                    .ADDR   F10D4   ;ADDRESS OF EXPECTED DATA TABLE
  (1)  4678  03                                        .BYTE   3       ;PACKING MODE 3
  (1)  4679  04                                        .BYTE   4       ;BYTE COUNT THIS XFR
  (1)  467A  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
```

```
1948  467B                                  XFRP    F10HC1,4,1,2
 (1)  467B  EF   46                                 .ADDR   F10HC1  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  467D  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  467E  01                                      .BYTE   1       ;BYTE COUNT THIS XFR
 (1)  467F  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1949  4680                                  XFRP    F10HC2,4,2,2
 (1)  4680  FB   46                                 .ADDR   F10HC2  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4682  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4683  02                                      .BYTE   2       ;BYTE COUNT THIS XFR
 (1)  4684  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1950  4685                                  XFRP    F10HC3,4,3,2
 (1)  4685  07   47                                 .ADDR   F10HC3  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4687  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4688  03                                      .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  4689  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1951  468A                                  XFRP    F10HC4,4,4,2
 (1)  468A  13   47                                 .ADDR   F10HC4  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  468C  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  468D  04                                      .BYTE   4       ;BYTE COUNT THIS XFR
 (1)  468E  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1952  468F                                  XFRP    F10HC5,4,5,4
 (1)  468F  1F   47                                 .ADDR   F10HC5  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4691  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4692  05                                      .BYTE   5       ;BYTE COUNT THIS XFR
 (1)  4693  04                                      .BYTE   4       ;4 SCLKS EXPECTED THIS XFR
1953  4694                                  XFRP    F10HC6,4,6,4
 (1)  4694  2B   47                                 .ADDR   F10HC6  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4696  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4697  06                                      .BYTE   6       ;BYTE COUNT THIS XFR
 (1)  4698  04                                      .BYTE   4       ;4 SCLKS EXPECTED THIS XFR
```

```
1955  4699                                    XFRP    F10HD1,6,1,2
 (1)  4699  EF   46                                     .ADDR   F10HD1  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  469B  06                                          .BYTE   6       ;PACKING MODE 6
 (1)  469C  01                                          .BYTE   1       ;BYTE COUNT THIS XFR
 (1)  469D  02                                          .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1956  469E                                    XFRP    F10HD2,6,2,2
 (1)  469E  FB   46                                     .ADDR   F10HD2  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46A0  06                                          .BYTE   6       ;PACKING MODE 6
 (1)  46A1  02                                          .BYTE   2       ;BYTE COUNT THIS XFR
 (1)  46A2  02                                          .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1957  46A3                                    XFRP    F10HD3,6,3,2
 (1)  46A3  07   47                                     .ADDR   F10HD3  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46A5  06                                          .BYTE   6       ;PACKING MODE 6
 (1)  46A6  03                                          .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  46A7  02                                          .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1958  46A8                                    XFRP    F10HD4,6,4,2
 (1)  46A8  13   47                                     .ADDR   F10HD4  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46AA  06                                          .BYTE   6       ;PACKING MODE 6
 (1)  46AB  04                                          .BYTE   4       ;BYTE COUNT THIS XFR
 (1)  46AC  02                                          .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1959  46AD                                    XFRP    F10HD5,6,5,2
 (1)  46AD  1F   47                                     .ADDR   F10HD5  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46AF  06                                          .BYTE   6       ;PACKING MODE 6
 (1)  46B0  05                                          .BYTE   5       ;BYTE COUNT THIS XFR
 (1)  46B1  02                                          .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1960  46B2                                    XFRP    F10HD6,6,6,4
 (1)  46B2  2B   47                                     .ADDR   F10HD6  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46B4  06                                          .BYTE   6       ;PACKING MODE 6
 (1)  46B5  06                                          .BYTE   6       ;BYTE COUNT THIS XFR
 (1)  46B6  04                                          .BYTE   4       ;4 SCLKS EXPECTED THIS XFR
1961  46B7  FF   FF                             .ADDR   @177777 ;END OF THE TABLE
1962                                    ;HERE IS THE EXPECTED DATA TABLES FOR ALL DATA TRANSFERS
1963
1964  46B9                              F11N:   ;11-NORMAL EXPECTED DATA
1965  46B9  04                                  .BYTE   @4      ;DDRA   3 BYTE XFR OF
1966  46BA  08                                  .BYTE   @10     ;DDRB   1,2,3,ZERO
1967  46BB  00                                  .BYTE   0
1968  46BC  0C                                  .BYTE   @14     ;2ND SCLK
1969  46BD  00                                  .BYTE   0
1970  46BE  00                                  .BYTE   0
1971
1972  46BF                              F15N:   ;15-NORMAL EXPECTED DATA
1973  46BF  08                                  .BYTE   @10     ;DDRA   3 BYTE XFR OF
1974  46C0  04                                  .BYTE   @4      ;DDRB   1,2,3,ZERO
1975  46C1  00                                  .BYTE   0
1976  46C2  00                                  .BYTE   0       ;2ND SCLK
1977  46C3  0C                                  .BYTE   @14
1978  46C4  00                                  .BYTE   0
1979
1980  46C5                              F10C1:  ;10-COMPAT BYTE COUNT -1
1981  46C5  00                                  .BYTE   0       ;DATA IS
1982  46C6  04                                  .BYTE   @4      ;       1,Z,Z,Z
1983  46C7  00                                  .BYTE   C
1984  46C8  00                                  .BYTE   0       ;2ND SCLK
```

```
1985  46C9  00                        .BYTE   0
1986  46CA  00                        .BYTE   0
1987
1988  46CB                    F10C2:  ;10-COMPAT BYTE COUNT =2
1989  46CB  08                        .BYTE   @10     ;DATA IS
1990  46CC  04                        .BYTE   @4      ;       1,2,Z,Z
1991  46CD  00                        .BYTE   0
1992  46CE  00                        .BYTE   0       ;2ND SCLK
1993  46CF  00                        .BYTE   0
1994  46D0  00                        .BYTE   0
1995
1996  46D1                    F10C3:  ;10-COMPAT BYTE COUNT =3
1997  46D1  08                        .BYTE   @10     ;DATA IS
1998  46D2  04                        .BYTE   @4      ;       1,2,3,Z
1999  46D3  00                        .BYTE   0
2000  46D4  00                        .BYTE   0       ;2ND SCLK
2001  46D5  C0                        .BYTE   @300
2002  46D6  00                        .BYTE   0
2003  46D7                    F10D1:  ;10-DUMP BYTE COUNT =1
2004  46D7  00                        .BYTE   0       ;DATA IS
2005  46D8  04                        .BYTE   @4      ;       1,Z,Z,Z,Z
2006  46D9  00                        .BYTE   0
2007  46DA  C0                        .BYTE   0       ;2ND SCLK
2008  46DB  00                        .BYTE   0
2009  46DC  00                        .BYTE   0
2010
2011  46DD                    F10D2:  ;10-DUMP BYTE COUNT -2
2012  46DD  08                        .BYTE   @10     ;DATA IS
2013  46DE  04                        .BYTE   @4      ;       1,2,Z,Z,Z
2014  46DF  00                        .BYTE   0
2015  46E0  00                        .BYTE   0       ;2ND SCLK
2016  46E1  00                        .BYTE   0
2017  46E2  00                        .BYTE   0
2018
2019  46E3                    F10D3:  ;10-DUMP BYTE COUNT =3
2020  46E3  08                        .BYTE   @10     ;DATA IS
2021  46E4  04                        .BYTE   @4      ;       1,2,3,Z,Z
2022  46E5  00                        .BYTE   0
2023  46E6  00                        .BYTE   0       ;2ND SCLK
2024  46E7  C0                        .BYTE   @300
2025  46E8  00                        .BYTE   0
2026
2027  46E9                    F10D4:  ;10-DUMP BYTE COUNT -4
2028  46E9  08                        .BYTE   @10     ;DATA IS
2029  46EA  04                        .BYTE   @4      ;       1,2,3,4,Z
2030  46EB  00                        .BYTE   0
2031  46EC  00                        .BYTE   0       ;2ND SCLK
2032  46ED  C1                        .BYTE   @301
2033  46EE  00                        .BYTE   0
2034
2035  46EF                    F10HD1:
2036  46EF                    F10HC1: ;10-HI-DEN-COMPAT BYTE COUNT -1
2037  46EF  00                        .BYTE   0       ;DATA IS
2038  46F0  04                        .BYTE   @4      ;       1,Z,Z,Z,Z,Z,Z,Z,Z
```

```
2039  46F1  00                              .BYTE   0
2040  46F2  00                              .BYTE   0       ;2ND SCLK
2041  46F3  00                              .BYTE   0
2042  46F4  00                              .BYTE   0
2043  46F5  00                              .BYTE   0       ;3RD SCLK
2044  46F6  00                              .BYTE   0
2045  46F7  00                              .BYTE   0
2046  46F8  00                              .BYTE   0       ;4TH SCLK
2047  46F9  00                              .BYTE   0
2048  46FA  00                              .BYTE   0
2049  46FB                          F10HD2:
2050  46FB                          F10HC2: ;10-HI-DEN-COMPAT BYTE COUNT =2
2051  46FB  08                              .BYTE   @10     ;DATA IS
2052  46FC  04                              .BYTE   @4      ;       1,2,Z,Z,Z,Z,Z,Z,Z,Z
2053  46FD  00                              .BYTE   0
2054  46FE  00                              .BYTE   0       ;2ND SCLK
2055  46FF  00                              .BYTE   0
2056  4700  00                              .BYTE   0
2057  4701  00                              .BYTE   0       ;3RD SCLK
2058  4702  00                              .BYTE   0
2059  4703  00                              .BYTE   0
2060  4704  00                              .BYTE   0       ;4TH SCLK
2061  4705  00                              .BYTE   0
2062  4706  00                              .BYTE   0
2063
2064  4707                          F10HD3:
2065  4707                          F10HC3: ;10-HI-DEN-COMPAT BYTE COUNT =3
2066  4707  08                              .BYTE   @10     ;DATA IS
2067  4708  04                              .BYTE   @4      ;       1,2,3,Z,Z,Z,Z,Z,Z,Z
2068  4709  00                              .BYTE   0
2069  470A  00                              .BYTE   0       ;2ND SCLK
2070  470B  C0                              .BYTE   @300
2071  470C  00                              .BYTE   0
2072  470D  00                              .BYTE   0       ;3RD SCLK
2073  470E  00                              .BYTE   0
2074  470F  00                              .BYTE   0
2075  4710  00                              .BYTE   0       ;4TH SCLK
2076  4711  00                              .BYTE   0
2077  4712  00                              .BYTE   0
2078
2079  4713                          F10HD4:
2080  4713                          F10HC4: ;10-HI-DEN-COMPAT BYTE COUNT -4
2081  4713  08                              .BYTE   @10     ;DATA IS
2082  4714  04                              .BYTE   @4      ;       1,2,3,4,Z,Z,Z,Z,Z,Z
2083  4715  00                              .BYTE   0
2084  4716  00                              .BYTE   0       ;2ND SCLK
2085  4717  C1                              .BYTE   @301
2086  4718  00                              .BYTE   0
2087  4719  00                              .BYTE   0       ;3RD SCLK
2088  471A  00                              .BYTE   0
2089  471B  00                              .BYTE   0
2090  471C  00                              .BYTE   0       ;4TH SCLK
2091  471D  00                              .BYTE   0
2092  471E  00                              .BYTE   0
```

```
2093  471F                          F10HD5:
2094  471F                          F10HC5: ;10-HI-DEN-COMPAT BYTE COUNT =5
2095  471F  08                             .BYTE   @10     ;DATA IS
2096  4720  04                             .BYTE   @4      ;       1,2,3,4,5,Z,Z,Z,Z
2097  4721  00                             .BYTE   0
2098  4722  00                             .BYTE   0       ;2ND SCLK
2099  4723  C1                             .BYTE   @301
2100  4724  00                             .BYTE   0
2101  4725  00                             .BYTE   0       ;3RD SCLK
2102  4726  40                             .BYTE   @100
2103  4727  01                             .BYTE   1
2104  4728  00                             .BYTE   0       ;4TH SCLK
2105  4729  00                             .BYTE   0
2106  472A  00                             .BYTE   0
2107
2108  472B                          F10HD6:
2109  472B                          F10HC6: ;10-HI-DEN-COMPAT BYTE COUNT =6
2110  472B  08                             .BYTE   @10     ;DATA IS
2111  472C  04                             .BYTE   @4      ;       1,2,3,4,5,6,Z,Z,Z
2112  472D  00                             .BYTE   0
2113  472E  00                             .BYTE   0       ;2ND SCLK
2114  472F  C1                             .BYTE   @301
2115  4730  C0                             .BYTE   0
2116  4731  80                             .BYTE   @200    ;3RD SCLK
2117  4732  41                             .BYTE   @101
2118  4733  01                             .BYTE   @1
2119  4734  00                             .BYTE   0       ;4TH SCLK
2120  4735  00                             .BYTE   0
2121  4736  00                             .BYTE   0
2122  4737                          F10HD7:
2123  4737                          F10HC7: ;10-HI-DEN-COMPAT BYTE COUNT =7
2124  4737  08                             .BYTE   @10     ;DATA IS
2125  4738  04                             .BYTE   @4      ;       1,2,3,4,5,6,7,Z,Z
2126  4739  00                             .BYTE   0
2127  473A  00                             .BYTE   0       ;2ND SCLK
2128  473B  C1                             .BYTE   @301
2129  473C  00                             .BYTE   0
2130  473D  81                             .BYTE   @201    ;3RD SCLK
2131  473E  41                             .BYTE   @101
2132  473F  01                             .BYTE   @1
2133  4740  00                             .BYTE   0       ;4TH SCLK
2134  4741  00                             .BYTE   0
2135  4742  00                             .BYTE   0
2136
2137  4743                          F10HD8:
2138  4743                          F10HC8: ;10-HI-DEN-COMPAT BYTE COUNT =8
2139  4743  08                             .BYTE   @10     ;DATA IS
2140  4744  04                             .BYTE   @4      ;       1,2,3,4,5,6,7,10,Z
2141  4745  00                             .BYTE   0
2142  4746  00                             .BYTE   0       ;2ND SCLK
2143  4747  C1                             .BYTE   @301
2144  4748  00                             .BYTE   0
2145  4749  81                             .BYTE   @201    ;3RD SCLK
2146  474A  41                             .BYTE   @101
```

```
 2147  474B  01                              .BYTE   @1
 2148  474C  00                              .BYTE   @0      ;4TH SCLK
 2149  474D  20                              .BYTE   @40
 2150  474E  00                              .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                          ********************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ********************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ********************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ********************************************************************************
                          12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                          12.0          PUSH    B
                           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
                          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
                           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                           4.0          ORA     B               ;ADD IN THE SELECTED PORT #
                          10.0          OUT     MBSEL           ;RESET TO THIS PORT
                           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                           4.0          XRA     A               ;CLEAR TU COMMAND A
                          10.0          OUT     TCMD            ;
                           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                           7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SLLECT'' COMMAND
                          10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
                           4.0          XRA     A
                          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
                          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
                           4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                           4.0          MOV     A,B
                           7.0          CPI     4               ;DONE?
                          10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                          10.0          POP     B               ;RESET B & C
                          10.0          POP     PSW             ;ALL DONE
                          10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
                                SSUB
                                ;*********************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 9
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    DECREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 0
                                ;*    ENDDO
                                ;*ENDSUB
                                S
                                ;*********************************************************************

 477A F5              12.0  T4X5:   PUSH    PSW             ;SAVE REGISTER A + PSW
 477B C5              12.0          PUSH    B               ;SAVE REGISTER B + C
 477C D5              12.0          PUSH    D               ;SAVE REGISTER D + E
 477D E5              12.0          PUSH    H               ;SAVE REGISTER H + L
                                ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                                ;CLEAR THE TRANSLATED DATA TABLE.
 477E 06 0A            7.0          MVI     B,10            ;SET UP LOOP COUNT
 4780 11 10 48        10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
 4783 AF               4.0          XRA     A               ;CLEAR A
 4784 12               7.0  D4X5:   STAX    D               ;STORE IN THE TABLE
 4785 13               6.0          INX     D               ;UPDATE TABLE POINTER
 4786 05               4.0          DCR     B               ;DECREMENT LOOP COUNT
 4787 C2 84 47        10.0          JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
 478A 0E 09            7.0          MVI     C,9             ;SET UP TRACK COUNT
```

```
2240  478C  06  04        7.0  B4X5:  MVI   B,4          ;SET UP BIT COUNT
2241  478E  11  0C  48    10.0         LXI   D,TRNIN
2242  4791  AF             4.0         XRA   A            ;CLEAR THE GROUP POSITION COUNT
2243  4792  32  1A  48    13.0         STA   GP4X5
2244  4795  1A             7.0  C4X5:  LDAX  D            ;GET A DATA BYTE
2245  4796  A7             4.0         ANA   A            ;SET CONDITION BITS
2246  4797  E2  9B  47    10.0         JPO   PO4X5        ;ODD PARITY LEAVE CARRY CLEAR
2247  479A  37             4.0         STC                ;EVEN PARITY SET CARRY
2248  479B  1F             4.0  PO4X5: RAR                ;SHIFT OUT DESIRED BIT
2249  479C  12             7.0         STAX  D            ;STORE RESULT BACK IN TEMP TABLE
2250  479D  3A  1A  48    13.0         LDA   GP4X5        ;GET THE GROUP POSITION BYTE
2251  47A0  17             4.0         RAL                ;PUT IN THIS DATA BIT
2252  47A1  32  1A  48    13.0         STA   GP4X5        ;SAVE THE UPDATED GROUP POSITION BYTE
2253  47A4  13             6.0         INX   D            ;UPDATE THE TABLE POINTER
2254  47A5  05             4.0         DCR   B            ;DECREMENT THE BIT COUNT
2255  47A6  C2  95  47    10.0         JNZ   C4X5         ;DO UNTIL ALL 4 BITS ARE RECEIVED
2256
2257                                ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2258
2259                                ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2260  47A9  06  05        7.0         MVI   B,5          ;SET UP LOOP COUNT
2261  47AB  11  10  48    10.0         LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2262  47AE  13             6.0  E4X5:  INX   D            ;POINT TO PARITY BIT
2263  47AF  1A             7.0         LDAX  D            ;GET PARITY BIT
2264  47B0  A7             4.0         ANA   A            ;CLEAR THE CARRY BIT
2265  47B1  1F             4.0         RAR                ;SHIFT TO THE CARRY BIT
2266  47B2  12             7.0         STAX  D            ;STORE IT BACK
2267  47B3  1B             6.0         DCX   D            ;DECREMENT TO DATA BITS
2268  47B4  1A             7.0         LDAX  D            ;GET THE DATA BITS
2269  47B5  1F             4.0         RAR                ;SHIFT IN CARRY BITS
2270  47B6  12             7.0         STAX  D            ;STORE BACK IN TABLE
2271  47B7  13             6.0         INX   D            ;POINT TO NEXT TABLE ENTRY
2272  47B8  13             6.0         INX   D            ;
2273  47B9  05             4.0         DCR   B            ;DECREMENT LOOP COUNT
2274  47BA  C2  AE  47    10.0         JNZ   E4X5         ;DO UNTIL LOOP COUNT=0
2275  47BD  11  EA  47    10.0         LXI   D,TAB4X5     ;LOAD ADDRESS OF TRANSLATION TABLE
2276  47C0  26  00        7.0         MVI   H,0          ;CLEAR REGISTER H
2277  47C2  3A  1A  48    13.0         LDA   GP4X5        ;GET GROUP POSITION COUNT
2278  47C5  6F             4.0         MOV   L,A
2279  47C6  19            10.0         DAD   D            ;ADD GROUP POSITION COUNT TO TABLE
2280  47C7  7E             7.0         MOV   A,M          ;GET TRANSLATED DATA
2281  47C8  32  1B  48    13.0         STA   TRNTMP
2282                                ;''TRNTMP'' = THE TRANSLATED DATA
2283
2284  47CB  11  10  48    10.0         LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2285  47CE  06  05        7.0         MVI   B,5          ;SET UP LOOP COUNT
2286  47D0  3A  1B  48    13.0  F4X5:  LDA   TRNTMP       ;GET TRANSLATED DATA
2287  47D3  13             6.0         INX   D            ;POINT TO PARITY POSITION
2288  47D4  A7             4.0         ANA   A            ;CLEAR CARRY BIT
2289  47D5  1F             4.0         RAR                ;SHIFT OUT A BIT
2290  47D6  32  1B  48    13.0         STA   TRNTMP       ;STORE TRANSLATED DATA
2291  47D9  1A             7.0         LDAX  D            ;GET THE PARITY BIT (ALWAYS ZERO)
2292  47DA  17             4.0         RAL                ;ROLL IN THE CARRY BIT
2293  47DB  12             7.0         STAX  D            ;STORE AS THE PARITY BIT
```

```
2294  47DC  13          6.0   INX   D      ;UPDATE TABLE POINTER
2295  47DD  05          4.0   DCR   B      ;DECREMENT THE LOOP COUNT
2296  47DE  C2  D0  47  10.0   JNZ   F4X5   ;DO UNTIL LOOP COUNT-0
2297
2298                           ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2299
2300  47E1  0D          4.0   DCR   C      ;DECREMENT THE TRACK COUNT
2301  47E2  C2  8C  47  10.0   JNZ   B4X5   ;DO UNTIL ALL TRACKS TRANSLATED
2302  47E5  E1          10.0   POP   H      ;RESTORE REGISTER H + L
2303  47E6  D1          10.0   POP   D      ;RESTORE REGISTER D + E
2304  47E7  C1          10.0   POP   B      ;RESTORE REGISTER B + C
2305  47E8  F1          10.0   POP   PSW    ;RESTORE REGISTER A + PSW
2306  47E9  C9          10.0   RET          ;RETURN TO USER
```

```
2308                                    .SBTTL  TABLE 4 X 5 TRANSLATION
2309
2310                                    ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2311                                    ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2312                                    ;   1 2 3 4 / 5 6 7 8                  10 9 8 7 6 / 5 4 3 2 1
2313                                    ;
2314                                    ;       0000                               10011
2315                                    ;       0001                               11011
2316                                    ;       0010                               01001
2317                                    ;       0011                               11001
2318                                    ;       0100                               10111
2319                                    ;       0101                               10101
2320                                    ;       0110                               01101
2321                                    ;       0111                               11101
2322                                    ;       1000                               01011
2323                                    ;       1001                               10010
2324                                    ;       1010                               01010
2325                                    ;       1011                               11010
2326                                    ;       1100                               01111
2327                                    ;       1101                               10110
2328                                    ;       1110                               01110
2329                                    ;       1111                               11110
2330
2331  47EA  13                          TAB4X5: .BYTE   @23
2332  47EB  1B                                  .BYTE   @33
2333  47EC  09                                  .BYTE   @11
2334  47ED  19                                  .BYTE   @31
2335  47EE  17                                  .BYTE   @27
2336  47EF  15                                  .BYTE   @25
2337  47F0  0D                                  .BYTE   @15
2338  47F1  1D                                  .BYTE   @35
2339  47F2  0B                                  .BYTE   @13
2340  47F3  12                                  .BYTE   @22
2341  47F4  0A                                  .BYTE   @12
2342  47F5  1A                                  .BYTE   @32
2343  47F6  0F                                  .BYTE   @17
2344  47F7  16                                  .BYTE   @26
2345  47F8  0E                                  .BYTE   @16
2346  47F9  1E                                  .BYTE   @36
```

```
                                .SBTTL  SUBROUTINE VARIABLES

                        SDDRA:  .BYTE   0               ;STORAGE FOR DDR A DATA
                        SDDRB:  .BYTE   0               ;STORAGE FOR DDR B DATA
                        SDDRC:  .BYTE   0               ;STORAGE FOR DDR C DATA
                        SCLKCT: .BYTE   0               ;COUNT OF 'SCLK' RECEIVED
                        ESCLK:  .BYTE   0               ;NUMBER OF EXPECTED SCLK PER PASS
                        FMTNUM: .BYTE   0               ;SELECTED PACKING MODE PER PASS
                        BCNT:   .BYTE   0               ;NUMBER OF BYTES IN CURRENT XFR
                        EXPADR: .ADDR   0               ;ADDRESS OF THE EXPECTED DATA TABLE
                        UNITMP: .BYTE   0               ;PORT NUMBER TO USE AS RESPONDED BY HOST
                        SUNIT:  .BYTE   0               ;STORAGE OF THE SELECTED UNIT NUMBER
                        CCTLWD: .BYTE   0               ;LOAD INTO THE SYSTEM CLOCK CONTROL WORD
                        HAVEXFR: .BYTE  0               ;FLAG TO INDICATE IF 'XFER' EVER SEEN
                        ONCE:   .BYTE   0               ;DATA XFR ONCE FLAG ... FOR GRONKING THE WMC

                        PTABLE: .ADDR   0               ;ADDRESS OF CURRENT XFR PARAMETER TABLE
                        NPTABLE: .ADDR  0               ;ADDRESS OF NEXT PTABLE

                        TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
                                                        ;BE TRANSLATED
                        TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
                                                        ;TRANSLATION
                        GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
                                                        ;BE TRANSLATED
                        TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
```

```
                                        .SBTTL  SUBROUTINE CLEAR ECC
                                        $
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
                                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
                                        $
                                        ; ************************************************************************

                                  4.0   CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
                                 13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
                                 10.0          RET                     ;RETURN TO USER
```

```
2386                                    .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
2387  4821                              S
 (1)                                    ; ************************************************************************
2388                                    ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
2389                                    ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
2390                                    ;TO THE ANSI STANDARD ECC POLYNOMIAL.
2391  4821                              S
 (1)                                    ; ************************************************************************
2392
2393  4821  E5              12.0  ECC:    PUSH    H               ;SAVE THE ADDRESS POINTER IN H & L
2394  4822  C5              12.0          PUSH    B               ;SAVE B & C
2395  4823  21  48  48      10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
2396  4826  AE               7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
2397  4827  5F               4.0          MOV     E,A             ;SAVE XOR RESULT IN E
2398  4828  E6  10           7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
2399  482A  7B               4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
2400  482B  CA  30  48       10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
2401  482E  EE  23            7.0          XRI     $23             ;ELSE-XOR WITH 23
2402  4830  5F               4.0  ECC1:    MOV     E,A             ;STORE THE ECC RESULT IN E
2403  4831  AF               4.0          XRA     A               ;CLEAR A
2404  4832  4F               4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
2405  4833  21  40  48       10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
2406  4836  CD  49  48       18.0          CALL    TRANS           ;TRANSLATE THE BITS
2407  4839  79               4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
2408  483A  32  48  48       13.0          STA     ECCCHR          ;STORE RESULT
2409  483D  C1               10.0          POP     B
2410  483E  E1               10.0          POP     H               ;RESET H & L
2411  483F  C9               10.0          RET
2412
2413  4840  08                     ECCTBL: $08                     ;BIT 0 = POSITION 3
2414  4841  20                             $20                     ;BIT 1 = POSITION 5
2415  4842  02                             $02                     ;BIT 2 = POSITION 1
2416  4843  40                             $40                     ;BIT 3 = POSITION 6
2417  4844  80                             $80                     ;BIT 4 = POSITION 7
2418  4845  01                             $01                     ;BIT 5 = POSITION 0
2419  4846  10                             $10                     ;BIT 6 = POSITION 4
2420  4847  04                             $04                     ;BIT 7 = POSITION 2
2421
2422  4848  00                     ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;
4849  06  01              7.0   TRANS:  MVI     B,1             ;INIT 'B' TO BIT POSITION 0
484B  7B                  4.0   TRANS1: MOV     A,E             ;GET CHAR TO BE TRANSLATED
484C  A0                  4.0           ANA     B               ;SEE IF BIT POSITION IN 'B' IS SET
484D  CA  53  48         10.0           JZ      TRANS2          ;DO NEXT BIT POSITION IF NOT SET
4850  79                  4.0           MOV     A,C             ;GET PREVIOUS RESULT OF 'OR'
4851  B6                  7.0           ORA     M               ;'OR' IN NEW POSITION
4852  4F                  4.0           MOV     C,A             ;SAVE RESULT

4853  78                  4.0   TRANS2: MOV     A,B
4854  07                  4.0           RLC                     ;POSITION MASK TO NEXT BIT
4855  47                  4.0           MOV     B,A
4856  D8                 12.0           RC                      ;EXIT WHEN ALL POSITIONS DONE
4857  23                  6.0           INX     H               ;POINT TO NEXT TABLE ENTRY
4858  C3  4B  48         10.0           JMP     TRANS1          ;PROCESS NEXT BIT

485B                                    $
 (1)                                    ; ***************************************************************************
                                        ;CLKSYS - ROUTINE TO CLOCK THE SYSTEM USING THE CONTENTS OF 'CCTLWD'
                                        ;       AS A BASE AND TOGGLING THE CLOCK BIT IN THE CLOCK CONTROL WORD
485B                                    $
 (1)                                    ; ***************************************************************************

485B  F5                 12.0   CLKSYS: PUSH    PSW
485C  3A  05  48         13.0           LDA     CCTLWD          ;GET THE CONTROL WORD BASE
485F  F6  40              7.0           ORI     SSCLK           ;GET THE CLOCK BIT
4861  D3  F0             10.0           OUT     CLKCTL          ;SET THE CLOCK ON
4863  E6  BF              7.0           ANI     @277            ;CLEAR THE CLOCK BIT
4865  D3  F0             10.0           OUT     CLKCTL          ;FINISH THE CLOCK CYCLE
4867  F1                 10.0           POP     PSW
4868  C9                 10.0           RET
      0000                      .END
```

```
A      -%0007    ADATA  = 0094    AMTIEP= 0001    AMTIE7= 0002
ARAIDF= 0098    ASAVE   4F9B    ATTCD   4F97    AXNUM   4F91
B      =%0000    BADST  = 0090    BCNT    4800    BIT0   = 0001
BIT1   = 0002    BIT15  = 8000    BIT2   = 0004    BIT3   = 0008
BIT4   = 0010    BIT5   = 0020    BIT6   = 0040    BIT7   = 0080
BIT8   = 0100    BIT9   = 0200    BRKPBC= 4F0A    BRKRAM= 4F10
BRKSTR= 4E60    BRKXCT= 4F00    BSAVE   4F9C    BYTCNT= 00D4
BYTEH   4F24    BYTEL   4F23    B4X5    478C    C      =%0001
CASCT   4F21    CASCTL= 00A0    CASSTA= 00A0    CATTH  = 0089
CATTL  = 0088    CBUSST= 00A1    CBYTH  = 008B    CBYTL  = 008A
CCTLWD  4805    CDG1H  = 0087    CDG1L  = 0086    CDG2H  = 0093
CDG2L  = 0092    CDG3H  = 0095    CDG3L  = 0094    CDVTH  = 008D
CDVTL  = 008C    CHPTIE= 0028    CH0TIE= 0020    CH1TIE= 0021
CH2TIE= 0022    CH3TIE= 0023    CH4TIE= 0024    CH5TIE= 0025
CH6TIE= 0026    CH7TIE= 0027    CKLOP  = 2815    CLEAR   474F
CLECC   481C    CLKCTL= 00F0    CLKRS   4419    CLKSYS  485B
CLOCK   4F26    CLRLP   4753    CMC0H  = 0099    CMC0L  = 0098
CMC1H  = 009B    CMC1L  = 009A    CMC2H  = 009D    CMC2L  = 009C
CMC3H  = 009F    CMC3L  = 009E    CMINH  = 0097    CMINL  = 0096
CNTCTL= C0D7    CNV1    4618    CONVRT  4612    CRCECC  464D
CRCGRP  4646    CRCG2   464A    CRCWRD= 0018    CSAVE   4F9D
CSRLH  = 0091    CSRLL  = 0090    CTCH   = 0085    CTCL   = 0084
CTSTH  = 008F    CTSTL  = 008E    CXCTH  = 0081    CXCTL  = 0080
CXINH  = 0083    CXINL  = 0082    C.     = 0001    C.AVAI= 0080
C.DP   = 0008    C.DSE  = 0010    C.DTU  = 0003    C.DVA  = 0008
C.FAIL= 00FC    C.FMT  = 0070    C.FNCT= 003E    C.GO   = 0001
C.INTC= 00FE    C.MAIN= 0020    C.NSA  = 0080    C.RCT  = 00FC
C.SER  = 0080    C.SHR  = 0040    C.SKPC= 000F    C.TAPE= 0040
C.WCS  = 0002    C1EC1   4424    C1EC1L  442C    C1EC2   443B
C1EC2L  4443    C4X5    4795    D      =%0002    DATACT= 00D0
DBUS    4F28    DBUSCT= 00C0    DBUSST= 00C0    DDRA   = 00D8
DDRAIN= 0010    DDRB   = 00D9    DDRBIN= 0002    DDRC   = 00DA
DDRCIN= 0001    DDRCO  = 0088    DDRCTL= 00DB    DIAFLG  4F22
DIAGPG= 4300    DIAGRM= 4F90    DIARD  = 000B    DONE1  = 0045
DONINT= 0010    DSAVE   4F9E    DSE    = 0006    DUMM1   4507  G
DUMM1A  44FC  G  DUMM2   4543  G  D.ATH0= 0001    D.ATH1= 0002
D.EOTD= 0010    D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004
D.TACH= 0008    D.WR4  = 0080    D4X5    4784    E      =%0003
ECC     4821    ECCBAD= 0042    ECCCHR  4848    ECCCOR= 0019
ECCOK  = 0041    ECCSTA= 001A    ECCTBL  4840    ECCTST= 000E
ECC1    4830    EDATA  = 0095    EOTCLR= 0003    ERFLG   4F93
ERLP   = 2809    ERLPA  = 280F    ERLPB  = 2812    ERLPE  = 280C
ERNUM   4F90    ERRCNT= 00D6    ESAVE   4F9F    ESCLK   47FE
EXIT    45F8    EXPADR  4801    E.ACRC= 0010    E.AMT  = 0020
E.CDP  = 0080    E.CRC  = 0080    E.PNTR= 0008    E.RPE  = 0040
E.STEC= 0001    E.TTEC= 0002    E.UNC  = 0004    E4X5    47AE
FIFORD= 006A    FMTNUM  47FF    FORMAT  4F25    FOUND   4327
FPORT   4347    FWDTST= 0061    F10C1   46C5    F10C2   46CB
F10C3   46D1    F10D1   46D7    F10D2   46DD    F10D3   46E3
F10D4   46E9    F10HC1  46EF    F10HC2  46FB    F10HC3  4707
F10HC4  4713    F10HC5  471F    F10HC6  472B    F10HC7  4737
F10HC8  4743    F10HD1  46EF    F10HD2  46FB    F10HD3  4707
F10HD4  4713    F10HD5  471F    F10HD6  472B    F10HD7  4737
F10HD8  4743    F11N    46B9    F15N    46BF    F4X5    47D0
```

```
GCRID = 0089        GCRSET= 0002        GOODTM= 0092        GP4X5   481A
H     =%0004        HAVEXF  4806        HLSAVE  4FA0        IE    - 0008
INTSTA= 00E0        ITERA   4F9A        I.PWR = 0020        I.RMPE= 0040
I5.5  = 0010        I6.5  = 0020        I7.5  = 0040        KCALL = 005F
KCLR  = 007B        KDEP  = 003F        KENAB = 0078        KEXAM = 003E
KEYBRD= 00C8        KEY1  = 0078        KEY10 = 006D        KEY11 = 006E
KEY12 = 006F        KEY13 = 005C        KEY14 = 005D        KEY15 = 005E
KEY16 = 005F        KEY17 = 003C        KEY18 = 003D        KEY19 = 003E
KEY2  = 0079        KEY20 = 003F        KEY3  = 007A        KEY4  = 007B
KEY5  = 0074        KEY6  = 0075        KEY7  = 0076        KEY8  = 0077
KEY9  = 006C        KINTA = 006F        KLDAD = 003D        KN0   = 003C
KN1   = 005C        KN2   = 005D        KN3   = 005E        KN4   = 006C
KN5   = 006D        KN6   = 006E        KN7   = 0074        KN8   = 0075
KN9   = 0076        KU2   = 0079        KU3   = 007A        KU8   = 0077
L     =%0005        LBLANK= 000F        LCE   = 000B        LCH   = 000C
LCL   = 000D        LCP   = 000E        LC0   = 0000        LC1   = 0001
LC2   = 0002        LC3   = 0003        LC4   = 0004        LC5   = 0005
LC6   - 0006        LC7   = 0007        LC8   = 0008        LC9   = 0009
LDFIFO  462C        LDLEDA= 00CA        LDLEDB= 00CB        LDLEDC= 00CC
LDLEDD= C0CD        LDLEDE= 00CE        LDLEDF= 00CF        LKDIAG= 2800
LKKBD - 004C        LKKEY = 0049        LKLWMG= 0058        LKLWMP= 0055
LKLWPG= 0052        LKLWPP= 004F        LKMOD7= 0046        LKOPR = 0046
LOADIT  4627        LPFLG   4F94        LPNUM   4F92        M     =%0006
MBSEL = 00E0        MB.A  = 0008        MB.B  = 0004        MEMTOP= 4FFF
MINUS = 000A        MM    - 8000        MSE   = 0008        MSGN  = 0007
MTACLR= 0000        MT.ARA= 0020        MT.CPE= 0080        MT.DSE= 0001
MT.FWD= 0040        MT.INH= 0008        MT.LWR= 0004        MT.MOT= 0002
MT.NWT= 0080        MT.PEC= 0040        MT.PSB= 0004        MT.PS0= 0001
MT.PS1- 0002        MT.REV= 0020        MT.WRT= 0010        MT.Z  = 0008
M.ATA = 0080        M.CAPE= 0020        M.CONT= 0080        M.DEM = 0020
M.EBL - 0010        M.EXC = 0008        M.FAIL= 0008        M.ILR = 0010
M.INIT= 0010        M.OCC = 0020        M.ONLI= 0001        M.PE  = 0040
M.PORT= 0080        M.RDEN= 0002        M.RDPE= 0008        M.RUN = 0004
M.SCLK= 0001        M.TRA = 0040        M.UNIT= 0007        M.WCLK= 0040
M.WCLN= 0080        M.WREN= 0080        M5.5  = 0001        M6.5  = 0002
M7.5  = 0004        NOTCAP= 0088        NOUNIT  431E  G     NPTABL  480A
OKAY  = 00FF        ONCE    4807        OPRRAM= 4300        OPSTRT= 0058
OPVER = 0040        PADCNT= 00D5        PADCRC= 0080        PDIAG = 0048
PEID  = 008A        PENAB = 004C        PESET = 0001        PL    = 00B1
P04X5   479B        PRDD  = 004C        PRENF = 009C        PS    = 00B2
PSTAT = 0048        PSW   =%0009        PTABLE  4808        P.AMTP= 0001
P.BCTC= 0040        P.CMDP= 0020        P.INTE= 0080        P.LCS = 0040
P.LWR - 0020        P.RDP = 0002        P.RPEN= 0004        P.RPST= 0002
P.RPOE- 0020        P.RP1E= 0010        P.RP2E= 0020        P.RP3E= 0010
P.SING= 0080        P.STAT= 0002        P.STPE= 0080        P.TACH= 0008
P.TUPR- 0010        P.WCSP= 0004        P.WDS = 0040        P.WFLP= 0001
P.WPEN- 0010        P.WPOE= 0008        P.WP1E= 0004        P.WP2E= 0008
P.WP3E- 0004        P.5VOK= 0002        QUE   = 281B        QUEM  = 281E
RAMT  - 0010        RARA  = 0006        RARAI = 0004        RCHBD0= 0048
RCHBD1= 0047        RCHOK - 0046        RCHRWD  464C        RCHTST= 000C
RCLRT - 000D        RCMD  = 000B        RCMLP = 0003        RCONT = 0080
RDATA = 0017        RDCLK = 0010        RDON  = 0011        READG = 0007
REND  - 0014        REQST = 2806        RESCHR= 00D1        REVTST= 0064
REWIND  0004        RFIFOL- 0008        RGCLK = 0002        RGCRI - 0003
```

```
RGECC   4645          RGRP    463E          RGRP2   4642          RIBG  = 0001
RILL  = 0012          RINST = 000C          RMCTST= 0008          RMK2  = 0013
RNOP  = 0000          RPATH = 0001          RPBAD = 0044          RPCHI = 0001
RPCLK = 0003          RPCTL = 0009          RPEI  = 0002          RPFAIL= 0000
RPF1  = 009D          RPF2  = 009E          RPOK  = 0043          RPOSTN= 0016
RPSTA = C015          RRCMT = 000A          RSTAT = 0002          RTIEB = 000A
RTIER = 0030          RTM   = 0005          RUNKI = 0009          RUPTST= 005E
RWDUNL= 0005          R.AMT = 0001          R.BOP = 0008          R.DATA= 0040
R.DON = 0002          R.DRDY= 0010          R.END = 0010          R.ILL = 0004
R.JVOK= 0004          R.MK2 = 0008          R.PLOD= 0008          R.PLOO= 0010
R.PLO1= 0020          R.POST= 0020          R.STNM= 0002          R.STOP= 0004
R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040          R.VOK = 0080
R00H  = 0081          R00L  = 0080          R01H  = 0083          R01L  = 0082
R02H  = 0085          R02L  = 0084          R03H  = 0087          R03L  = 0086
R04H  = 0089          R04L  = 0088          R05H  = 008B          R05L  = 008A
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = C09D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SCLKCT  47FD          SDDRA   47FA          SDDRB   47FB
SDDRC   47FC          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          SUNIT   4804          TAB4X5  47EA          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST1   4300          TMF   = 0099          TMRDY = 0040          TRANS   4849
TRANS1  484B          TRANS2  4853          TRKENA= 00D2          TRNIN   480C
TRNOUT  4810          TRNTMP  481B          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR- 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002          T1CKA   4573
T1CKAC  459C  G       T1CKB   45A2          T1CKBC  45B6  G       T1CKC   45BC
T1CKCC  45D0  G       T1CLK   44CB          T1CLKA  44C1          T1CLK1  44DB
T1CLK2  44E5          T1CLK3  44EC          T1EMK   4468          T1EMKL  446D
T1ERNS  4502          T1FOK   43CF          T1LOOP  436D          T1MK2   44A0
T1MK2L  44A5          T1M2L2  44B4          T1PLO   4452          T1RCLR  43B5
T1RFIL  43C0          T1SCKE  453E          T1SCKL  4513          T1SCK0  4549
T1SCLK  450D          T1SG1   447C          T1SG2   4485          T1SG3   448E
T1SG4   4497          T1ST    4422          T4X5    477A          UIBG  = 00A1
UNITMP  4803          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          XFRPTB  464E          X.DONN= 0080          X.ENAB- 0040
```

```
X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001            Y      -%000B
.     = 4869


 ERRORS DETECTED:  0

 *WMC7.A78/PTP,WMC7=NLIST,PARAM,MACRO,LIST,WMC7
 RUN-TIME: 4 6 0 SECONDS
 CORE USED:  10K
```

```
                                .TITLE  WMC8 - BYTE ASSEMBLY/SKIP COUNT OF READ PATH DATA
                                .SBTTL  TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
                                ;ID     WMC8-WRITE MICRO CONTROLLER PART #8

4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WMC BYTE ASSEMBLY/SKIP COUNT TEST FROM READ FORWARD DATA
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
                                ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
                                ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
                                ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
                                ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
                                ;*TO 10-HI DEN COMPAT) IN A READ FORWARD DATA TRANSFER. THE TEST WILL
                                ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
                                ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
                                ;*EXPECTED. THE READ PATH WILL BE FILLED WITH 8 SUBGROUPS WORTH OF DATA
                                ;*REGUARDLESS OF THE AMOUNT OF EXPECTED DATA GROUPS TO BE TESTED (USUALLY
                                ;*4 SUBGROUPS OF DATA WILL BE SUFFICIENT TO TEST A PACKING MODE). THE EXTRA
                                ;*SUBGROUPS OF DATA INJECTED WILL ELIMINATE THE NECESSITY OF CALCULATING
                                ;*THE 'SPECIAL' CHARACTERS FROM THE TAPE IN GCR MODE (I.E. CRC, ACRC, &
                                ;*RESID CHARACTERS). EVERY 4 BYTES OF DATA TO BE INJECTED WILL BE TURNED
                                ;*INTO A 5 BYTE SUBGROUP TO SIMULATE DATA FROM THE 'TAPE'. THE 'ECC'
                                ;*WILL BE CALCULATED FOR EACH 7 BYTE OF DATA INJECTED AND ALSO INJECTED
                                ;*AT THE PROPER TIME INTO THE FIFO'S.
4300                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  REQUEST A TU PORT FROM 'HOST' CPU
                                ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
                                ;*  :  THEN-REPORT OPERATOR ERROR
                                ;*  :  ELSE-PROCEED
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  CLEAR ALL PORTS
                                ;*  :  INIT THE SYSTEM
                                ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
                                ;*  :  INIT TRANSFER TABLE PARAMETER POINTER - PTABLE
                                ;*  :  LOAD SELECTED SKIP COUNT & PACKING MODE
                                ;*  :  INIT COUNTERS + CONTROLS
                                ;*  :  LOAD BYTE COUNTER + PAD COUNTER
                                ;*  :  RESTART WRITE MICROCONTROLLER
                                ;*  :  CLOCK WMC THROUGH RESTART CODE
                                ;*  :  START 'GCR' READ DATA XFR
                                ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
                                ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  :  CLOCK THE FIFO'S
```

```
                                ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
                                ;*  :  WAIT
                                ;*  :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*  :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
                                ;*  :  SETUP A WATCHDOG TIMER
                                ;*  :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
                                ;*  :  IF WATCHDOG TIMER=0
                                ;*  :  :  THEN-REPORT NO 'SCLK'
                                ;*  :  :  ELSE-CONT
                                ;*  :  ENDIF
                                ;*  :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
                                ;*  :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
                                ;*  :  IF 18 BIT DATA SAME AS EXPECTED
                                ;*  :  :  THEN-PROCEED
                                ;*  :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
                                ;*  :  ENDIF
                                ;*  :  POINT TO NEXT PARAMETER TABLE ENTRY SET
                                ;*  :  IF ALL DONE WITH DATA TRANSFERS IN 'PTABLE'
                                ;*  :  :  THEN - DECREMENT THE ITERATION COUNTER
                                ;*  :  :  :     IF ALL ITERATIONS HAVE BEEN EXECUTED
                                ;*  :  :  :     :  THEN-EXIT
                                ;*  :  :  :     :  ELSE-CONTINUE
                                ;*  :  :  :     ENDIF
                                ;*  :  :  :  ELSE-CONTINUE
                                ;*  :  :  ENDIF
                                ;*  :  :  DO UNTIL ALL ITERATIONS EXECUTED
                                ;*  ENDDO
                                ;*ENDTST
1402  4300                      SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1403                            ;*WMC8 MICRO TEST 01
1404                            ;*WMC8 MICRO ERROR 01
1405                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1406                            ;*M8959 (WMC) & ALL OTHER BOARDS
1407                            ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1408                            ;*FATAL ERROR - TEST ABORTED
1409                            ;*
1410                            ;*WMC8 MICRO TEST 01
1411                            ;*WMC8 MICRO ERROR 02
1412                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1413                            ;*M8959 (WMC) & ALL OTHER BOARDS
1414                            ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
1415                            ;*AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
1416                            ;*FATAL ERROR - TEST ABORTED
1417                            ;*
1418                            ;*WMC8 MICRO TEST 01
1419                            ;*WMC8 MICRO ERROR 03
1420                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1421                            ;*M8959 (WMC) & ALL OTHER BOARDS
1422                            ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1423                            ;*DURING A READ FORWARD COMMAND
1424                            ;*FATAL ERROR - TEST ABORTED
```

```
                                        ;*
                                        ;*WMC8 MICRO TEST 01
                                        ;*WMC8 MICRO ERROR 04
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC8 MICRO TEST 01
                                        ;*WMC8 MICRO ERROR 05
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC8 MICRO TEST 01
                                        ;*WMC8 MICRO ERROR 06
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
 (1)                                    ; ***************************************************************************
                                        TEST1:  TESTX   @1
4300                                                    MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4300  3E  01            7.0                       CALL    TSET            ;SETUP THE TEST
 (1)  4302  CD  03  28       18.0
                                        ;%WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;&M8959 (WMC) & ALL OTHER BOARDS
                                        ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING

                                                REQ     7
 (1)  4305  CD  06  28       18.0                       CALL    REQST
 (1)  4308  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (1)  4309  00  00                                      .WORD                   ;SYSTEM '-' COUNT
 (1)  430B  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                          .BYTE                   ;DATA COMPARE FLAG IF -1
 (1)  430E  07                                          .BYTE   7               ;REQUEST CODE
430F                                            RIN     R12L                    ;GET THE SELECTED PORT FROM 'HOST'
 (1)  430F  DB  94           10.0                       IN      R12L                    ;READ R12L INTO AC
 (1)  4311  7F                4.0                       MOV     A,A                     ;RETRY LINK
4312  32  5E  47              13.0                  STA     UNITMP          ;SAVE IT
4315  A7                       4.0                  ANA     A               ;SET THE CONDITION BITS
4316  C2  27  43              10.0                  JNZ     FOUND           ;SEE IF ANY WERE SELECTED IN 'HOST'
```

```
1462  4319                                        ERR     EXIT,NOUNIT
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                          .BYTE
 (1)  431E  CD  15  28      18.0                NOUNIT::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  4F  45      10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1463                                        ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
1464                                        ;>FATAL ERROR - TEST ABORTED
1465  4324  C3  4F  45      10.0                JMP     EXIT
1466                                        ;HERE IF THE HOST CPU INDICATED A PORT TO USE
1467
1468  4327  CD  AA  46      18.0          FOUND:  CALL    CLEAR           ;INIT ALL THE PORTS IN THE PORT BOARD
1469  432A  3A  5E  47      13.0                  LDA     UNITMP          ;GET THE RESPONSE FROM 'HOST' CPU
1470  432D  06  00           7.0                 MVI     B,0             ;INIT 'B'
1471  432F  E6  01           7.0                 ANI     1               ;IS UNIT 0 SELECTED?
1472  4331  C2  47  43      10.0                 JNZ     FPORT           ;YUP - FOUND THE PORT TO USE
1473  4334  04               4.0                 INR     B               ;NO - COUNT THE PORT
1474  4335  3A  5E  47      13.0                 LDA     UNITMP          ;GET RESPONSE AGAIN
1475  4338  E6  02           7.0                 ANI     2               ;PORT 1 SELECTED?
1476  433A  C2  47  43      10.0                 JNZ     FPORT           ;YUP
1477  433D  04               4.0                 INR     B               ;COUNT THE PORT
1478  433E  3A  5E  47      13.0                 LDA     UNITMP          ;GET RESPONSE
1479  4341  E6  04           7.0                 ANI     4               ;PORT 2 SELECTED?
1480  4343  C2  47  43      10.0                 JNZ     FPORT           ;YUP
1481  4346  04               4.0                 INR     B               ;NO - ASSUME PORT 3
1482
1483  4347  78               4.0          FPORT:  MOV     A,B             ;PUT UNIT # IN AC
1484  4348  32  5F  47      13.0                 STA     SUNIT           ;SAVE THE UNIT UNDER TEST
1485  434B  DB  E0          10.0                 IN      INTSTA          ;GET THE PORT SELECT STATUS
1486  434D  E6  80           7.0                 ANI     BIT7
1487  434F  B0               4.0                 ORA     B               ;ADD IN THE SELECTED PORT #
1488  4350  D3  E0          10.0                 OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
1489  4352  3E  60           7.0                 MVI     A,P.LWR.P.LCS   ;GET THE LOOP-COMMAND-STATUS BIT
1490  4354  D3  48          10.0                 OUT     PDIAG           ;SELECT IT
1491  4356  AF               4.0                 XRA     A
1492  4357  D3  44          10.0                 OUT     TAMT            ;CLEAR THE AMTIE LINES
1493  4359  D3  D2          10.0                 OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS
1494  435B  D3  D2          10.0                 OUT     TRKENA
1495  435D  3E  04           7.0                 MVI     A,P.RPEN        ;SET READ PATH ENABLE
1496  435F  D3  4C          10.0                 OUT     PENAB           ;NOW
1497  4361  3E  10           7.0                 MVI     A,RDCLK         ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1498  4363  D3  F0          10.0                 OUT     CLKCTL
1499  4365  D3  C0          10.0                 OUT     DBUSCTL         ;CLEAR THE DATA BUS CONTROL WORD
1500  4367  21  A5  45      10.0                 LXI     H,XFRPTB        ;POINT TO THE 1ST DATA XFR PARAMETER TABLE
1501  436A  22  24  46      16.0                 SHLD    PTABLE          ;SAVE THE TABLE ADDRESS
1502                                        ;INIT THE COUNTER CONTROL AND COUNTERS
1503
1504  436D                                T1LOOP: INICNT
 (1)  436D  3E  30           7.0                        MVI     A,@060          ;INIT THE
 (1)  436F  D3  D7          10.0                        OUT     CNTCTL          ;BYTE COUNTER
 (1)  4371  3E  70           7.0                        MVI     A,@160          ;INIT THE
 (1)  4373  D3  D7          10.0                        OUT     CNTCTL          ;PAD COUNTER
```

```
 (1)  4375  3E  B0        7.0                    MVI   A,@260          ;INIT THE
 (1)  4377  D3  D7       10.0                    OUT   CNTCTL          ;ERROR COUNTER
1505  4379                             CLRECT
 (1)  4379  AF            4.0                    XRA   A               ;CLEAR THE ACCUMULATOR
 (1)  437A  D3  D6       10.0                    OUT   ERRCNT          ;CLEAR BITS 7-0
 (1)  437C  D3  D6       10.0                    OUT   ERRCNT          ;CLEAR BITS 15-8
1506
1507  437E  2A  24  46    16.0          LHLD    PTABLE                ;GET THE PARAMETER POINTER
1508  4381  7E             7.0          MOV     A,M                   ;GET THE LOW BYTE OF EXPECTED ADDRESS
1509  4382  32  5C  47    13.0          STA     EXPADR                ;SAVE IT
1510  4385  23             6.0          INX     H                     ;POINT TO THE HIGH BYTE OF ADDRESS
1511  4386  7E             7.0          MOV     A,M                   ;GET THE HIGH BYTE
1512  4387  32  5D  47    13.0          STA     EXPADR+1              ;SAVE IT
1513  438A  23             6.0          INX     H                     ;POINT TO THE PACKING MODE NUMBER
1514  438B  7E             7.0          MOV     A,M                   ;GET IT
1515  438C  32  5A  47    13.0          STA     FMTNUM                ;SAVE IT
1516  438F  23             6.0          INX     H                     ;POINT TO THE SKIP COUNT FOR THIS XFR
1517  4390  7E             7.0          MOV     A,M                   ;GET IT
1518  4391  32  5B  47    13.0          STA     SKIPCT                ;SAVE IT
1519  4394  23             6.0          INX     H                     ;POINT TO THE # SCLK'S TO CHECK IN XFR
1520  4395  7E             7.0          MOV     A,M                   ;GET THE COUNT
1521  4396  32  59  47    13.0          STA     ESCLK                 ;SAVE THE NUMBER
1522  4399  23             6.0          INX     H                     ;POINT TO THE 1ST WORD IN NEXT ENTRY
1523  439A  22  24  46    16.0          SHLD    PTABLE                ;SAVE THE ADDRESS OF NEW TABLE
1524
1525  439D  3A  5A  47    13.0          LDA     FMTNUM                ;GET THE PACKING MODE # AGAIN
1526  43A0  07             4.0          RLC
1527  43A1  07             4.0          RLC
1528  43A2  07             4.0          RLC
1529  43A3  07             4.0          RLC
1530  43A4  E6  70         7.0          ANI     $70                   ;POSITION THE FMT #
1531  43A6  47             4.0          MOV     B,A                   ;TEMP SAVE
1532  43A7  3A  5B  47    13.0          LDA     SKIPCT                ;GET THE SELECTED SKIP COUNT FOR XFR
1533  43AA  B0             4.0          ORA     B                     ;ADD IN THE FORMAT NUMBER
1534  43AB  D3  D0        10.0          OUT     DATACTL               ;LOAD THE DATA CONTROL WORD
1535  43AD                             ROUT    R02H                  ;SAVE FOR ERROR PRINTING
 (1)  43AD  D3  85        10.0                  OUT     R02H                  ;WRITE AC INTO R02H
 (1)  43AF  7F             4.0                  MOV     A,A                   ;RETRY LINK
1536  43B0  3E  9B         7.0          MVI     A,@233                ;SET THE DDR CONTROL TO 'IN'
1537  43B2  D3  DB        10.0          OUT     DDRCTL                ;TO READ THE 'DDR'
1538  43B4  3E  00         7.0          MVI     A,@0                  ;GET THE RESIDUAL CHAR WORD
1539  43B6  D3  D1        10.0          OUT     RESCHR                ;LOAD IT
1540  43B8  3E  32         7.0          MVI     A,50                  ;SETUP THE BYTE COUNTER FOR 8 DATA GROUPS
1541  43BA  D3  D4        10.0          OUT     BYTCNT
1542  43BC  AF             4.0          XRA     A
1543  43BD  D3  D4        10.0          OUT     BYTCNT
1544  43BF  3E  06         7.0          MVI     A,6                   ;NEED 6 PAD CHARACTERS FOR THE XFR
1545  43C1  D3  D5        10.0          OUT     PADCNT                ;LOAD IT
1546  43C3  AF             4.0          XRA     A
1547  43C4  D3  D5        10.0          OUT     PADCNT
1548  43C6                             ROUT    R02L                  ;CLEAR OUT THE REST OF REG 2 IN CAS
 (1)  43C6  D3  84        10.0                  OUT     R02L                  ;WRITE AC INTO R02L
 (1)  43C8  7F             4.0                  MOV     A,A                   ;RETRY LINK
1549  43C9                             ROUT    R05L                  ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
```

```
 (1)  43C9  D3  8A          10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  43CB  7F               4.0                    MOV     A,A             ;RETRY LINK
1550  43CC  32  58  47      13.0            STA     SCLKCT          ;INIT THE SCLK DETECTED COUNTER
1551
1552                                  ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1553
1554  43CF  3E  01            7.0            MVI     A,W.RST         ;GET THE WMC RESTART BIT
1555  43D1  D3  D3           10.0            OUT     WMCCTL          ;SET RESTART
1556  43D3  3E  15            7.0            MVI     A,SSTEP.RDCLK   ;GET SINGLE STEP FOR WMC
1557  43D5  D3  F0           10.0            OUT     CLKCTL          ;SET SINGLE STEP MODE
1558  43D7  32  60  47       13.0            STA     CCTLWD          ;SAVE THE WORD
1559  43DA  CD  B0  47       18.0            CALL    CLKSYS          ;CLOCK THE SYSTEM TO CAUSE RESTART TO
1560  43DD  CD  B0  47       18.0            CALL    CLKSYS          ;TO TAKE HOLD
1561  43E0  AF                4.0            XRA     A
1562  43E1  D3  D3           10.0            OUT     WMCCTL          ;FINISH THE RESTART CYCLE
1563
1564                                  ;CLOCK THE WMC THRU ITS RESTART AREA
1565
1566  43E3  06  B4            7.0            MVI     B,180           ;GET A LOOP COUNT
1567  43E5  CD  B0  47       18.0    CLKRS:  CALL    CLKSYS          ;CLOCK THE SYSTEM
1568  43E8  05                4.0            DCR     B               ;DECREMENT THE LOOP COUNTER
1569  43E9  C2  E5  43       10.0            JNZ     CLKRS           ;CLOCK TILL COUNT RUN OUT
1570
1571  43EC  3E  90            7.0            MVI     A,W.GCR.W.ENAB  ;SETUP FOR A GCR DATA XFR
1572  43EE  D3  D3           10.0    T1ST:   OUT     WMCCTL          ;LOAD THE WORD IN WMC
1573
1574                                  ;CALCULATE THE 1ST DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1575
1576  43F0  21  95  45       10.0    C1EC1:  LXI     H,DG1           ;POINT THE THE 1ST DATA GROUP DATA TABLE
1577  43F3  06  07            7.0            MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1578  43F5  CD  71  47       18.0            CALL    CLECC           ;INIT THE ECC CHARACTER
1579
1580  43F8  7E                7.0    C1EC1L: MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
1581  43F9  CD  76  47       18.0            CALL    ECC             ;CALC ECC WITH THIS CHARACTER
1582  43FC  23                6.0            INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1583  43FD  05                4.0            DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
1584  43FE  C2  F8  43       10.0            JNZ     C1EC1L          ;STAY HERE TILL ALL DATA BYTES USED
1585  4401  3A  9D  47       13.0            LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1586  4404  32  9C  45       13.0            STA     DG1ECC          ;SAVE THE CALCULATED ECC CHARACTER
1587
1588                                  ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1589
1590  4407  21  9D  45       10.0    C1EC2:  LXI     H,DG2           ;POINT THE THE 2ND DATA GROUP DATA TABLE
1591  440A  06  07            7.0            MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1592  440C  CD  71  47       18.0            CALL    CLECC           ;INIT THE ECC CHARACTER
1593
1594  440F  7E                7.0    C1EC2L: MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
1595  4410  CD  76  47       18.0            CALL    ECC             ;CALC ECC WITH THIS CHARACTER
1596  4413  23                6.0            INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1597  4414  05                4.0            DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
1598  4415  C2  0F  44       10.0            JNZ     C1EC2L          ;STAY HERE TILL ALL DATA BYTES USED
1599  4418  3A  9D  47       13.0            LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1600  441B  32  A4  45       13.0            STA     DG2ECC          ;SAVE THE CALCULATED ECC CHARACTER
1601  441E  3E  A8            7.0    T1PLO.  MVI     A,R.PLO1.R.TBJN!R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
```

```
1602                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1603  4420  D3  09        10.0          OUT     RPCTL
1604  4422  AF             4.0          XRA     A
1605  4423  D3  0A        10.0          OUT     RTIEB           ;CLEAR THE TIE BUS
1606  4425  D3  08        10.0          OUT     RFIFOL          ;CLOCK THE FIFO'S
1607  4427  3E  0D         7.0          MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1608  4429  D3  0B        10.0          OUT     RCMD
1609  442B  00             4.0          NOP                     ;WAIT
1610  442C  00             4.0          NOP                     ;WAIT
1611  442D  00             4.0          NOP
1612  442E  00             4.0          NOP
1613  442F  00             4.0          NOP
1614  4430  3E  A9         7.0          MVI     A,R.PLO1!R.STPC!R.PLOD.R.TBJN   ;STOP THE READ PATH
1615                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1616  4432  D3  09        10.0          OUT     RPCTL
1617
1618  4434  0E  03         7.0          MVI     C,@3            ;INIT THE LOOP COUNT FOR 6 DATA GROUPS
1619
1620  4436  21  95  45    10.0  T1SG1:  LXI     H,DG1           ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
1621  4439  CD  69  45    18.0          CALL    CONVRT          ;CONVERT SUBGROUP - 4X5
1622  443C  CD  7E  45    18.0          CALL    LOADIT          ;INJECT THE DATA INTO THE FIFO
1623
1624  443F  21  99  45    10.0  T1SG2:  LXI     H,SG2           ;POINT TO THE 2ND SUBGROUP OF DATA
1625  4442  CD  69  45    18.0          CALL    CONVRT          ;MAKE IT 4X5
1626  4445  CD  7E  45    18.0          CALL    LOADIT          ;INJECT IT INTO FIFO
1627
1628  4448  21  9D  45    10.0  T1SG3:  LXI     H,DG2           ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
1629  444B  CD  69  45    18.0          CALL    CONVRT          ;MAKE IT 4X5
1630  444E  CD  7E  45    18.0          CALL    LOADIT          ;INJECT THE DATA INTO FIFO
1631
1632  4451  21  A1  45    10.0  T1SG4:  LXI     H,SG4           ;POINT TO THE 4TH SUBGROUP OF DATA
1633  4454  CD  69  45    18.0          CALL    CONVRT          ;MAKE IT 4X5
1634  4457  CD  7E  45    18.0          CALL    LOADIT          ;INJECT THE DATA INTO FIFO
1635
1636  445A  0D             4.0          DCR     C               ;DECREMENT THE LOOP COUNTER
1637  445B  C2  36  44    10.0          JNZ     T1SG1           ;KEEP INJECTING DATA TILL 6 GROUPS IN FIFO
1638
1639  445E  3E  0B         7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
1640  4460  D3  0B        10.0          OUT     RCMD            ;COMMAND
1641  4462  11  01  00    10.0  T1CLKA: LXI     D,1             ;SET WATCH DOG INCREMENT
1642  4465  21  68  C5    10.0          LXI     H,-15000        ;SET WATCH DOG COUNT TO 15000
1643  4468  D3  0C        10.0  T1CLK:  OUT     RINST           ;STEP THE READ PATH
1644  446A  CD  B0  47    18.0          CALL    CLKSYS          ;CLOCK THE WMC
1645  446D  DB  C0        10.0          IN      DBUSSTA         ;GET THE DATABUS STATUS
1646  446F  E6  02         7.0          ANI     T.SCLK          ;SEE IF SCLK UP - DATA READY
1647  4471  C2  86  44    10.0          JNZ     T1SCLK          ;JUMP IF SCLK UP
1648  4474  19            10.0          DAD     D
1649  4475  C2  68  44    10.0          JNC     T1CLK           ;STAY IN LOOP UNLESS TIMEOUT
1650
```

```
1652  4478                                          ERR     T1LOOP,DUMM1,3
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4478  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447B  02                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447C  03                                              .BYTE   3
 (1)  447D  CD  15  28        18.0                  DUMM1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4480  DA  6D  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1653                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
1654                                    ;>AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
1655                                    ;<FATAL ERROR - TEST ABORTED
1656  4483  C3  4F  45        10.0            JMP     EXIT
1657
1658                                    ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO -0
1659                                    ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
1660                                    ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDICATING THE DATA IS
1661                                    ;ON THE MASSBUS
1662
1663  4486  06  FF             7.0  T1SCLK: MVI     B,$FF           ;SETUP A DROP DEAD COUNT
1664  4488  D3  0C            10.0  T1SCKL: OUT     RINST           ;CLOCK THE READ PATH ONCE
1665  448A  CD  B0  47        18.0          CALL    CLKSYS          ;CLOCK THE WMC TO CLEAR 'SCLK'
1666  448D  DB  C0            10.0          IN      DBUSSTA         ;SEE IF 'SCLK' IS GONE
1667  448F  E6  02             7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
1668  4491  CA  A6  44        10.0          JZ      T1SCK0          ;JUMP IF 'SCLK' = 0
1669  4494  05                 4.0          DCR     B               ;DECREMENT THE COUNTER
1670  4495  C2  88  44        10.0          JNZ     T1SCKL          ;LOOP BACK TILL COUNTER =0
1671  4498                                  ERR     T1LOOP,DUMM2,3
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4498  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0003                                            MSGN            MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4498  03                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  449C  03                                              .BYTE   3
 (1)  449D  CD  15  28        18.0                  DUMM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A0  DA  6D  43        10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1672                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1673                                    ;>DURING A READ FORWARD COMMAND
1674                                    ;<FATAL ERROR - TEST ABORTED
1675  44A3  C3  4F  45        10.0            JMP     EXIT
1676                                    ;HERE WHEN 'SCLK' WAS SET - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
1677                                    ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
1678
1679  44A6  DB  D8            10.0  T1SCK0: IN      DDRA            ;GET THE DDR A DATA
1680  44A8  32  55  47        13.0          STA     SDDRA           ;SAVE IT
1681  44AB                                  ROUT    R16L            ;SAVE FOR PRINTING
 (1)  44AB  D3  9C            10.0                  OUT     R16L            ;WRITE AC INTO R16L
 (1)  44AD  7F                 4.0                  MOV     A,A             ;RETRY LINK
1682  44AE  DB  D9            10.0          IN      DDRB            ;GET DDR B DATA
1683  44B0  32  56  47        13.0          STA     SDDRB           ;SAVE IT
1684  44B3                                  ROUT    R16H            ;SAVE FOR ERROR REPORTING
 (1)  44B3  D3  9D            10.0                  OUT     R16H            ;WRITE AC INTO R16H
 (1)  44B5  7F                 4.0                  MOV     A,A             ;RETRY LINK
1685  44B6  DB  DA            10.0          IN      DDRC            ;GET DDR C DATA
1686  44B8  E6  03             7.0          ANI     3
1687  44BA  32  57  47        13.0          STA     SDDRC           ;SAVE IT
```

```
1688  44BD  F6  80        7.0        ORI   BIT7          ;ADD IN THE PRINT FLAG
1689  44BF                           ROUT  R17L          ;SAVE IT FOR PRINTING
 (1)  44BF  D3  9E       10.0              OUT   R17L          ;WRITE AC INTO R17L
 (1)  44C1  7F            4.0              MOV   A,A           ;RETRY LINK
1690
1691  44C2  3A  58  47   13.0        LDA   SCLKCT        ;GET THE CURRENT SCLK COUNT
1692  44C5  3C            4.0        INR   A             ;COUNT THIS SCLK
1693  44C6  32  58  47   13.0        STA   SCLKCT        ;SAVE THE NEW COUNT
1694  44C9                           ROUT  R05L          ;ALSO IN CAS FOR ERROR REPORTING
 (1)  44C9  D3  8A       10.0              OUT   R05L          ;WRITE AC INTO R05L
 (1)  44CB  7F            4.0              MOV   A,A           ;RETRY LINK
1695  44CC  AF            4.0        XRA   A
1696  44CD                           ROUT  R05H
 (1)  44CD  D3  8B       10.0              OUT   R05H          ;WRITE AC INTO R05H
 (1)  44CF  7F            4.0              MOV   A,A           ;RETRY LINK
1697
1698                                ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED
1699
1700  44D0  2A  5C  47   16.0  T1CKA:  LHLD  EXPADR         ;POINT TO THE TABLE OF EXPECTED DATA
1701  44D3  7E            7.0        MOV   A,M           ;GET BITS 7-0
1702  44D4                           ROUT  R14L          ;SAVE FOR ERROR REPORTING
 (1)  44D4  D3  98       10.0              OUT   R14L          ;WRITE AC INTO R14L
 (1)  44D6  7F            4.0              MOV   A,A           ;RETRY LINK
1703  44D7  23            6.0        INX   H             ;POINT TO NEXT BYTE
1704  44D8  7E            7.0        MOV   A,M           ;GET BITS 15-8
1705  44D9                           ROUT  R14H          ;SAVE IT
 (1)  44D9  D3  99       10.0              OUT   R14H          ;WRITE AC INTO R14H
 (1)  44DB  7F            4.0              MOV   A,A           ;RETRY LINK
1706  44DC  23            6.0        INX   H
1707  44DD  7E            7.0        MOV   A,M           ;GET BITS 17-16
1708  44DE  F6  80         7.0        ORI   BIT7          ;ADD IN THE PRINT FLAG
1709  44E0                           ROUT  R15L          ;SAVE FOR ERROR REPORT
 (1)  44E0  D3  9A       10.0              OUT   R15L          ;WRITE AC INTO R15L
 (1)  44E2  7F            4.0              MOV   A,A           ;RETRY LINK
1710  44E3  3A  55  47   13.0        LDA   SDDRA         ;GET THE DDR A DATA
1711  44E6                           ROUT  ADATA         ;SAVE IN CASE OF ERROR DETECTED
 (1)  44E6  D3  94       10.0              OUT   ADATA         ;WRITE AC INTO ADATA
 (1)  44E8  7F            4.0              MOV   A,A           ;RETRY LINK
1712  44E9  2A  5C  47   16.0        LHLD  EXPADR        ;POINT TO THE EXPECTED DATA TABLE
1713  44EC  BE            7.0        CMP   M             ;ACTUAL DATA = EXPECTED?
1714  44ED  CA  F9  44   10.0        JZ    T1CKAC        ;JUMP IF OK
1715  44F0  7E            7.0        MOV   A,M           ;GET THE EXPECTED DATA
1716  44F1                           ROUT  EDATA         ;SAVE FOR ERROR REPORT
 (1)  44F1  D3  95       10.0              OUT   EDATA         ;WRITE AC INTO EDATA
 (1)  44F3  7F            4.0              MOV   A,A           ;RETRY LINK
1717  44F4                     ERRB  T1LOOP,T1CKAC,@10
 (1)                           ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  44F4  CD  12  28   18.0                    CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                                 MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F7  04                                   .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44F8  08                                   .BYTE @10             ;PRINT ROUTINE NUMBER
 (1)  44F9  CD  15  28   18.0          T1CKAC::        CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FC  DA  6D  43   10.0                    JC    T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1718                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
```

```
1719  44FF  23                    6.0  T1CKB:  INX   H                ;POINT TO THE EXPECTED DATA FOR DDR B
1720  4500  3A  56  47             13.0          LDA   SDDRB            ;GET THE DDR B DATA
1721  4503                               ROUT  ADATA                   ;SAVE FOR ERROR REPORT
 (1)  4503  D3  94                 10.0          OUT   ADATA                 ;WRITE AC INTO ADATA
 (1)  4505  7F                      4.0          MOV   A,A                   ;RETRY LINK
1722  4506  BE                    7.0          CMP   M                  ;ACTUAL = EXPECTED DATA?
1723  4507  CA  13  45             10.0          JZ    T1CKBC             ;JUMP IF OK
1724  450A  7E                      7.0          MOV   A,M                ;GET THE EXPECTED DATA
1725  450B                               ROUT  EDATA                   ;SAVE FOR ERROR REPORT
 (1)  450B  D3  95                 10.0          OUT   EDATA                 ;WRITE AC INTO EDATA
 (1)  450D  7F                      4.0          MOV   A,A                   ;RETRY LINK
1726  450E                               ERRB  T1LOOP,T1CKBC,@10
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  450E  CD  12  28             18.0                CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                                       MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4511  05                                         .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4512  08                                         .BYTE @10              ;PRINT ROUTINE NUMBER
 (1)  4513  CD  15  28             18.0          T1CKBC::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4516  DA  6D  43             10.0                JC    T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1727                               ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
1728
1729  4519  23                    6.0  T1CKC:  INX   H                ;POINT TO EXPECTED DDR C DATA
1730  451A  3A  57  47             13.0          LDA   SDDRC            ;GET DDR C DATA
1731  451D                               ROUT  ADATA                   ;SAVE FOR ERROR REPORT
 (1)  451D  D3  94                 10.0          OUT   ADATA                 ;WRITE AC INTO ADATA
 (1)  451F  7F                      4.0          MOV   A,A                   ;RETRY LINK
1732  4520  BE                    7.0          CMP   M                  ;ACTUAL DATA = EXPECTED DDRC DATA?
1733  4521  CA  2D  45             10.0          JZ    T1CKCC             ;JUMP IF OK
1734  4524  7E                      7.0          MOV   A,M                ;GET THE EXPECTED DATA
1735  4525                               ROUT  EDATA                   ;SAVE FOR ERROR REPORT
 (1)  4525  D3  95                 10.0          OUT   EDATA                 ;WRITE AC INTO EDATA
 (1)  4527  7F                      4.0          MOV   A,A                   ;RETRY LINK
1736  4528                               ERRB  T1LOOP,T1CKCC,@10
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4528  CD  12  28             18.0                CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                                       MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  452B  06                                         .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  452C  08                                         .BYTE @10              ;PRINT ROUTINE NUMBER
 (1)  452D  CD  15  28             18.0          T1CKCC::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4530  DA  6D  43             10.0                JC    T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1737                               ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
1738
1739  4533  2A  5C  47             16.0          LHLD  EXPADR           ;GET THE CURRENT EXPECTED DATA ADDR.
1740  4536  23                      6.0          INX   H
1741  4537  23                      6.0          INX   H                ;UPDATE THE EXPECTED ADDRESS POINTER
1742  4538  23                      6.0          INX   H                ;TO THE DATA
1743  4539  22  5C  47             16.0          SHLD  EXPADR           ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
1744  453C  3A  58  47             13.0          LDA   SCLKCT           ;GET THE SCLK COUNT
1745  453F  21  59  47             10.0          LXI   H,ESCLK          ;POINT TO THE EXPECTED SCLK COUNT
1746  4542  BE                    7.0          CMP   M                ;DONE WITH THIS TEST PASS?
1747  4543  C2  62  44             10.0          JNZ   T1CLKA           ;JUMP IF STILL EXPECTING MORE SCLKS
1748
1749                               ;HERE TO SEE IF ALL XFRS IN THE 'PTABLE' HAVE BEEN DONE
1750
```

```
1751  4546  2A  24  46   16.0          LHLD    PTABLE          ;POINT TO THE TABLE
1752  4549  7E               7.0       MOV     A,M             ;GET THE 1ST WORD OF AN ENTRY
1753  454A  FE  FF            7.0       CPI     @377            ;END OF THE TABLE?
1754  454C  C2  6D  43   10.0          JNZ     T1LOOP          ;NO - DO THIS ENTRY
1755
1756  454F                          EXIT:   ENDTST  FOUND
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  454F                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  454F  CD  06  28   18.0                       CALL    REQST
 (2)  4552  00                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  4553  00  00                                 .WORD                   ;SYSTEM '*' COUNT
 (2)  4555  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4557  00                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4558  07                                     .BYTE   7               ;REQUEST CODE
 (1)  4559  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  455C  3D               4.0       DCR     A               ;DOWNCOUNT
 (1)  455D  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  4560  F2  27  43   10.0          JP      FOUND           ;DO TEST UNTIL TILL = 0
1757  4563  CD  AA  46   18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
1758  4566  C3  18  28   10.0          JMP     TSTEND
1759                          ;HERE IS THE SUBROUTINES NECESSARY FOR CONVERTING SUBGROUPS AND LOADING
1760                          ;THEM INTO THE FIFO
1761
1762  4569                    S
 (1)                          ; ***************************************************************************
1763                          ;*CONVRT -- THIS SUBROUTINE WILL GET A SUBGROUP OF DATA AS POINTED TO BY
1764                          ;*      THE H & L REGISTERS AND CAUSE THEM TO BE TRANSLATED (4X5). THE
1765                          ;*      TRANSLATED DATA WILL BE IN A TABLE CALLED 'TRNOUT'.
1766  4569                    S
 (1)                          ; ***************************************************************************
1767
1768  4569  D5            12.0  CONVRT: PUSH    D               ;SAVE D & E
1769  456A  11  61  47    10.0          LXI     D,TRNIN         ;POINT TO THE TEMP STORAGE FOR DATA BYTES
1770  456D  06  04         7.0          MVI     B,4             ;SETUP A LOOP COUNTER FOR 4 BYTES
1771
1772  456F  7E              7.0  CNV1:   MOV     A,M             ;GET A BYTE OF DATA
1773  4570  EB              4.0          XCHG                    ;POINT TO THE STORAGE
1774  4571  77              7.0          MOV     M,A             ;SAVE DATA BYTE IN TEMP STORAGE
1775  4572  EB              4.0          XCHG                    ;RESET POINTER
1776  4573  23              6.0          INX     H               ;POINT TO NEXT DATA BYTE
1777  4574  13              6.0          INX     D               ;POINT TO NEXT TEMP STORAGE ADDRESS
1778  4575  05              4.0          DCR     B               ;UPDATE THE LOOP COUNT
1779  4576  C2  6F  45   10.0          JNZ     CNV1            ;LOOP TILL 4 BYTES IN TEMP STORAGE
1780  4579  CD  D5  46   18.0          CALL    T4X5            ;CONVERT THE 'SUBGROUP' OF DATA
1781  457C  D1              10.0       POP     D
1782  457D  C9              10.0       RET
1783  457E                    S
 (1)                          ; ***************************************************************************
1784                          ;*LOADIT -- THIS SUBROUTINE WILL TAKE THE TRANSLATED SUBGROUP (4X5)
1785                          ;*      DATA FROM TABLE 'TRNOUT' AND LOAD THE DATA INTO THE READ
1786                          ;*      PATH FIFO.
1787  457E                    S
 (1)                          ; ***************************************************************************
1788
```

```
1789  457E  21  65  47      10.0  LOADIT: LXI   H,TRNOUT        ;POINT TO THE TRANSLATED DATA TABLE
1790  4581  06  05              7.0          MVI   B,5            ;GET THE LOOP COUNT FOR SUBGROUP
1791
1792  4583  7E                  7.0  LDFIFO: MOV   A,M            ;GET A BYTE OF DATA
1793  4584  D3  40             10.0          OUT   TCMD           ;STORE IT IN THE CMD REG FOR LOADING
1794  4586  23                  6.0          INX   H              ;POINT TO THE PARITY BIT
1795  4587  7E                  7.0          MOV   A,M            ;GET THE PARITY BIT
1796  4588  07                  4.0          RLC
1797  4589  F6  60              7.0          ORI   P.LWR!P.LCS    ;ADD IN THE OTHER NECESSARY BITS
1798  458B  D3  48             10.0          OUT   PDIAG          ;LOAD INTO CONTROL REG FOR LOADING
1799  458D  D3  08             10.0          OUT   RFIFOL         ;STROBE THE DATA INTO FIFO
1800  458F  23                  6.0          INX   H              ;POINT TO THE NEXT SET OF DATA
1801  4590  05                  4.0          DCR   B              ;DECREMENT THE LOOP COUNTER
1802  4591  C2  83  45         10.0          JNZ   LDFIFO         ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
1803  4594  C9                 10.0          RET
1804                                   ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
1805                                   ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ FORWARD'
1806
1807  4595  01                  DG1:     .BYTE  @1       ;1ST BYTE IN THE 1ST SUBGROUP
1808  4596  02                           .BYTE  @2       ;1ST SUBGROUP HAS 4 DATA BYTES IN IT
1809  4597  03                           .BYTE  @3
1810  4598  04                           .BYTE  @4       ;LAST BYTE IN THE 1ST SUBGROUP
1811
1812  4599  05                  SG2:     .BYTE  @5       ;1ST BYTE IN 2ND SUBGROUP
1813  459A  06                           .BYTE  @6       ;2ND SUBGROUP HAS 3 BYTES IN IT TO FINISH THE DATA
1814  459B  07                           .BYTE  @7       ;DATA GROUP OF 7 DATA BYTES
1815  459C  00                  DG1ECC:  .BYTE  0        ;STORAGE FOR THE 1ST ECC BYTE (CALCULATED)
1816
1817  459D  08                  DG2:     .BYTE  @10      ;1ST BYTE OF THE 3RD SUBGROUP
1818  459E  09                           .BYTE  @11      ;3RD SUBGROUP HAS 4 DATA BYTES
1819  459F  0A                           .BYTE  @12
1820  45A0  0B                           .BYTE  @13      ;LAST DATA BYTE IN 3RD SUBGROUP
1821
1822  45A1  0C                  SG4:     .BYTE  @14      ;1ST BYTE IN THE 4TH SUBGROUP
1823  45A2  0D                           .BYTE  @15      ;THIS SUBGROUP HAS ONLY 3 DATA BYTES
1824  45A3  0E                           .BYTE  @16      ;LAST BYTE IN 4TH SUBGROUP
1825  45A4  00                  DG2ECC:  .BYTE  0        ;2ND DATA GROUP ECC BYTE STORAGE (CALCULATED)
1826                                   ;HERE IS THE PARAMETER TABLE FOR ALL DATA XFRS TO TEST THE SKIP COUNTS
1827                                   ;IN ALL DATA PACKING MODES
1828
1829  45A5                      XFRPTB:  XFRP   F11N,0,1,2
 (1)  45A5  26  46                        .ADDR  F11N     ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45A7  00                            .BYTE  0        ;PACKING MODE 0
 (1)  45A8  01                            .BYTE  1        ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45A9  02                            .BYTE  2        ;2 EXPECTED THIS XFR
1830  45AA                               XFRP   F15N,1,1,2
 (1)  45AA  2C  46                        .ADDR  F15N     ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45AC  01                            .BYTE  1        ;PACKING MODE 1
 (1)  45AD  01                            .BYTE  1        ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45AE  02                            .BYTE  2        ;2 EXPECTED THIS XFR
1831  45AF                               XFRP   F10C1,2,1,2
 (1)  45AF  32  46                        .ADDR  F10C1    ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45B1  02                            .BYTE  2        ;PACKING MODE 2
 (1)  45B2  01                            .BYTE  1        ;SKIP COUNT - 1 FOR THIS XFR
```

```
                                                        .BYTE   2       ;2 EXPECTED THIS XFR
1832  45B4                                      XFRP    F10C2,2,2,2
                                                        .ADDR   F10C2   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45B4  38  46                                      .BYTE   2       ;PACKING MODE 2
 (1)  45B6  02                                          .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
 (1)  45B7  02                                          .BYTE   2       ;2 EXPECTED THIS XFR
 (1)  45B8  02
1833  45B9                                      XFRP    F10C3,2,3,2
 (1)  45B9  3E  46                                      .ADDR   F10C3   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45BB  02                                          .BYTE   2       ;PACKING MODE 2
 (1)  45BC  03                                          .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
 (1)  45BD  02                                          .BYTE   2       ;2 EXPECTED THIS XFR
1834  45BE                                      XFRP    F10D1,3,1,2
 (1)  45BE  44  46                                      .ADDR   F10D1   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45C0  03                                          .BYTE   3       ;PACKING MODE 3
 (1)  45C1  01                                          .BYTE   1       ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45C2  02                                          .BYTE   2       ;2 EXPECTED THIS XFR
1835  45C3                                      XFRP    F10D2,3,2,2
 (1)  45C3  4A  46                                      .ADDR   F10D2   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45C5  03                                          .BYTE   3       ;PACKING MODE 3
 (1)  45C6  02                                          .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
 (1)  45C7  02                                          .BYTE   2       ;2 EXPECTED THIS XFR
1836  45C8                                      XFRP    F10D3,3,3,2
 (1)  45C8  50  46                                      .ADDR   F10D3   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45CA  03                                          .BYTE   3       ;PACKING MODE 3
 (1)  45CB  03                                          .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
 (1)  45CC  02                                          .BYTE   2       ;2 EXPECTED THIS XFR
1837  45CD                                      XFRP    F10D4,3,4,2
 (1)  45CD  56  46                                      .ADDR   F10D4   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45CF  03                                          .BYTE   3       ;PACKING MODE 3
 (1)  45D0  04                                          .BYTE   4       ;SKIP COUNT = 4 FOR THIS XFR
 (1)  45D1  02                                          .BYTE   2       ;2 EXPECTED THIS XFR
1838  45D2                                      XFRP    F10HC1,4,1,4
 (1)  45D2  5C  46                                      .ADDR   F10HC1  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45D4  04                                          .BYTE   4       ;PACKING MODE 4
 (1)  45D5  01                                          .BYTE   1       ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45D6  04                                          .BYTE   4       ;4 EXPECTED THIS XFR
1839  45D7                                      XFRP    F10HC2,4,2,4
 (1)  45D7  68  46                                      .ADDR   F10HC2  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45D9  04                                          .BYTE   4       ;PACKING MODE 4
 (1)  45DA  02                                          .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
 (1)  45DB  04                                          .BYTE   4       ;4 EXPECTED THIS XFR
1840  45DC                                      XFRP    F10HC3,4,3,4
 (1)  45DC  74  46                                      .ADDR   F10HC3  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45DE  04                                          .BYTE   4       ;PACKING MODE 4
 (1)  45DF  03                                          .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
 (1)  45E0  04                                          .BYTE   4       ;4 EXPECTED THIS XFR
1841  45E1                                      XFRP    F10HC4,4,4,4
 (1)  45E1  80  46                                      .ADDR   F10HC4  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45E3  04                                          .BYTE   4       ;PACKING MODE 4
 (1)  45E4  04                                          .BYTE   4       ;SKIP COUNT = 4 FOR THIS XFR
 (1)  45E5  04                                          .BYTE   4       ;4 EXPECTED THIS XFR
1842  45E6                                      XFRP    F10HC5,4,5,2
 (1)  45E6  92  46                                      .ADDR   F10HC5  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45E8  04                                          .BYTE   4       ;PACKING MODE 4
```

```
  (1)  45E9  05                                  .BYTE   5       ;SKIP COUNT = 5 FOR THIS XFR
  (1)  45EA  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1843   45EB                          XFRP    F10HC6,4,6,2
  (1)  45EB  98   46                             .ADDR   F10HC6  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  45ED  04                                  .BYTE   4       ;PACKING MODE 4
  (1)  45EE  06                                  .BYTE   6       ;SKIP COUNT = 6 FOR THIS XFR
  (1)  45EF  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1844   45F0                          XFRP    F10HC7,4,7,2
  (1)  45F0  9E   46                             .ADDR   F10HC7  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  45F2  04                                  .BYTE   4       ;PACKING MODE 4
  (1)  45F3  07                                  .BYTE   7       ;SKIP COUNT = 7 FOR THIS XFR
  (1)  45F4  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1845   45F5                          XFRP    F10HC8,4,8,2
  (1)  45F5  A4   46                             .ADDR   F10HC8  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  45F7  04                                  .BYTE   4       ;PACKING MODE 4
  (1)  45F8  08                                  .BYTE   8       ;SKIP COUNT = 8 FOR THIS XFR
  (1)  45F9  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1846   45FA                          XFRP    F10HD1,6,1,4
  (1)  45FA  5C   46                             .ADDR   F10HD1  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  45FC  06                                  .BYTE   6       ;PACKING MODE 6
  (1)  45FD  01                                  .BYTE   1       ;SKIP COUNT - 1 FOR THIS XFR
  (1)  45FE  04                                  .BYTE   4       ;4 EXPECTED THIS XFR
1847   45FF                          XFRP    F10HD2,6,2,4
  (1)  45FF  68   46                             .ADDR   F10HD2  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  4601  06                                  .BYTE   6       ;PACKING MODE 6
  (1)  4602  02                                  .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
  (1)  4603  04                                  .BYTE   4       ;4 EXPECTED THIS XFR
1848   4604                          XFRP    F10HD3,6,3,4
  (1)  4604  74   46                             .ADDR   F10HD3  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  4606  06                                  .BYTE   6       ;PACKING MODE 6
  (1)  4607  03                                  .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
  (1)  4608  04                                  .BYTE   4       ;4 EXPECTED THIS XFR
1849   4609                          XFRP    F10HD4,6,4,2
  (1)  4609  8C   46                             .ADDR   F10HD4  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  460B  06                                  .BYTE   6       ;PACKING MODE 6
  (1)  460C  04                                  .BYTE   4       ;SKIP COUNT = 4 FOR THIS XFR
  (1)  460D  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1850   460E                          XFRP    F10HD5,6,5,2
  (1)  460E  92   46                             .ADDR   F10HD5  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  4610  06                                  .BYTE   6       ;PACKING MODE 6
  (1)  4611  05                                  .BYTE   5       ;SKIP COUNT = 5 FOR THIS XFR
  (1)  4612  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1851   4613                          XFRP    F10HD6,6,6,2
  (1)  4613  98   46                             .ADDR   F10HD6  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  4615  06                                  .BYTE   6       ;PACKING MODE 6
  (1)  4616  06                                  .BYTE   6       ;SKIP COUNT = 6 FOR THIS XFR
  (1)  4617  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1852   4618                          XFRP    F10HD7,6,7,2
  (1)  4618  9E   46                             .ADDR   F10HD7  ;ADDRESS OF EXPECTED DATA TABLE
  (1)  461A  06                                  .BYTE   6       ;PACKING MODE 6
  (1)  461B  07                                  .BYTE   7       ;SKIP COUNT = 7 FOR THIS XFR
  (1)  461C  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1853   461D                          XFRP    F10HD8,6,8,2
  (1)  461D  A4   46                             .ADDR   F10HD8  ;ADDRESS OF EXPECTED DATA TABLE
```

```
(1)  461F  06                          .BYTE   6       ;PACKING MODE 6
(1)  4620  08                          .BYTE   8       ;SKIP COUNT = 8 FOR THIS XFR
(1)  4621  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1854 4622  FF   FF                     .ADDR   @177777 ;END OF THE TABLE
1855
1856 4624  00   00          PTABLE:    .ADDR   0
1857                        ;HERE IS THE EXPECTED DATA TABLES FOR ALL DATA TRANSFERS
1858
1859 4626                   F11N:      ;11-NORMAL EXPECTED DATA
1860 4626  00                          .BYTE   0       ;DDRA   4 BYTE XFR OF
1861 4627  04                          .BYTE   @4      ;DDRB   FILL,1,2,3
1862 4628  00                          .BYTE   0
1863 4629  08                          .BYTE   @10     ;2ND SCLK
1864 462A  0C                          .BYTE   @14
1865 462B  00                          .BYTE   0
1866
1867 462C                   F15N:      ;15-NORMAL EXPECTED DATA
1868 462C  04                          .BYTE   @4      ;DDRA   4 BYTE XFR OF
1869 462D  00                          .BYTE   0       ;DDRB   FILL,1,2,3
1870 462E  00                          .BYTE   0
1871 462F  0C                          .BYTE   @14     ;2ND SCLK
1872 4630  08                          .BYTE   @10
1873 4631  C0                          .BYTE   0
1874
1875 4632                   F10C1:     ;10-COMPAT SKIP COUNT =1
1876 4632  04                          .BYTE   @4      ;DATA IS
1877 4633  00                          .BYTE   0       ;       FILL,1,2,3
1878 4634  00                          .BYTE   0
1879 4635  C0                          .BYTE   @300    ;2ND SCLK
1880 4636  80                          .BYTE   @200
1881 4637  00                          .BYTE   0
1882
1883 4638                   F10C2:     ;10-COMPAT SKIP COUNT -2
1884 4638  00                          .BYTE   0       ;DATA IS
1885 4639  00                          .BYTE   0       ;       FILL,FILL,1,2
1886 463A  00                          .BYTE   0
1887 463B  80                          .BYTE   @200    ;2ND SCLK
1888 463C  40                          .BYTE   @100
1889 463D  00                          .BYTE   0
1890
1891 463E                   F10C3:     ;10-COMPAT SKIP COUNT  3
1892 463E  00                          .BYTE   0       ;DATA IS
1893 463F  00                          .BYTE   0       ;       FILL,FILL,FILL,1
1894 4640  00                          .BYTE   0
1895 4641  40                          .BYTE   @100    ;2ND SCLK
1896 4642  00                          .BYTE   0
1897 4643  00                          .BYTE   0
1898 4644                   F10D1:     ;10-DUMP SKIP COUNT =1
1899 4644  04                          .BYTE   @4      ;DATA IS
1900 4645  00                          .BYTE   0       ;       FILL,1,2,3,4
1901 4646  00                          .BYTE   0
1902 4647  D0                          .BYTE   @320    ;2ND SCLK
1903 4648  80                          .BYTE   @200
1904 4649  00                          .BYTE   0
```

```
1905
1906  464A                          F10D2:  ;10-DUMP SKIP COUNT =2
1907  464A  00                               .BYTE   0       ;DATA IS
1908  464B  00                               .BYTE   0       ;       FILL,FILL,1,2,3
1909  464C  00                               .BYTE   0
1910  464D  8C                               .BYTE   @214    ;2ND SCLK
1911  464E  40                               .BYTE   @100
1912  464F  00                               .BYTE   0
1913
1914  4650                          F10D3:  ;10-DUMP SKIP COUNT =3
1915  4650  00                               .BYTE   0       ;DATA IS
1916  4651  00                               .BYTE   0       ;       FILL,FILL,FILL,1,2
1917  4652  00                               .BYTE   0
1918  4653  48                               .BYTE   @110    ;2ND SCLK
1919  4654  00                               .BYTE   0
1920  4655  00                               .BYTE   0
1921
1922  4656                          F10D4:  ;10-DUMP SKIP COUNT =4
1923  4656  00                               .BYTE   0       ;DATA IS
1924  4657  00                               .BYTE   0       ;       FILL,FILL,FILL,FILL,1
1925  4658  00                               .BYTE   0
1926  4659  04                               .BYTE   @4      ;2ND SCLK
1927  465A  C0                               .BYTE   0
1928  465B  00                               .BYTE   0
1929
1930  465C                          F10HD1:
1931  465C                          F10HC1: ;10-HI-DEN-COMPAT SKIP COUNT =1
1932  465C  04                               .BYTE   @4      ,DATA IS
1933  465D  00                               .BYTE   0       ;       F,1,2,3,4,5,6,7,10
1934  465E  00                               .BYTE   0
1935  465F  C0                               .BYTE   @300    ;2ND SCLK
1936  4660  80                               .BYTE   @200
1937  4661  00                               .BYTE   0
1938  4662  41                               .BYTE   @101    ;3RD SCLK
1939  4663  01                               .BYTE   1
1940  4664  01                               .BYTE   1
1941  4665  22                               .BYTE   @42     ;4TH SCLK
1942  4666  1C                               .BYTE   @34
1943  4667  00                               .BYTE   0
1944  4668                          F10HD2:
1945  4668                          F10HC2: ;10-HI-DEN-COMPAT SKIP COUNT -2
1946  4668  00                               .BYTE   0       ;DATA IS
1947  4669  00                               .BYTE   0       ;       F,F,1,2,3,4,5,6,7
1948  466A  00                               .BYTE   0
1949  466B  80                               .BYTE   @200    ;2ND SCLK
1950  466C  40                               .BYTE   @100
1951  466D  C0                               .BYTE   0
1952  466E  01                               .BYTE   @1      ;3RD SCLK
1953  466F  C1                               .BYTE   @301
1954  4670  00                               .BYTE   0
1955  4671  1D                               .BYTE   @35     ;4TH SCLK
1956  4672  18                               .BYTE   @30
1957  4673  00                               .BYTE   0
1958
```

```
1959  4674
1960  4674
1961  4674  00          F10HD3:
1962  4675  00          F10HC3: ;10-HI-DEN-COMPAT SKIP COUNT =3
1963  4676  00                  .BYTE   0       ;DATA IS
1964  4677  40                  .BYTE   0       ;       F,F,F,1,2,3,4,5,6
1965  4678  00                  .BYTE   0
1966  4679  00                  .BYTE   @100    ;2ND SCLK
1967  467A  C1                  .BYTE   0
1968  467B  80                  .BYTE   0
1969  467C  00                  .BYTE   @301    ;3RD SCLK
1970  467D  18                  .BYTE   @200
1971  467E  14                  .BYTE   0
1972  467F  00                  .BYTE   @30     ;4TH SCLK
1973                            .BYTE   @24
1974  4680                      .BYTE   0
1975  4680  00          F10HC4: ;10-HI-DEN-COMPAT SKIP COUNT =4
1976  4681  00                  .BYTE   0       ;DATA IS
1977  4682  00                  .BYTE   0       ;       F,F,F,F,1,2,3,4,5
1978  4683  00                  .BYTE   0
1979  4684  00                  .BYTE   0       ;2ND SCLK
1980  4685  00                  .BYTE   0
1981  4686  80                  .BYTE   @200    ;3RD SCLK
1982  4687  40                  .BYTE   @100
1983  4688  00                  .BYTE   0
1984  4689  17                  .BYTE   @27     ;4TH SCLK
1985  468A  10                  .BYTE   @20
1986  468B  00                  .BYTE   0
1987
1988  468C              F10HD4: ;10-HI-DEN-DUMP SKIP COUNT =4
1989  468C  80                  .BYTE   @200    ;DATA IS
1990  468D  40                  .BYTE   @100    ;       F,F,F,F,1,2,3,4,5
1991  468E  00                  .BYTE   0       ;ONLY GET LAST 2 SCLKS
1992  468F  17                  .BYTE   @27     ;2ND SCLK
1993  4690  10                  .BYTE   @20
1994  4691  00                  .BYTE   0
1995  4692              F10HD5:
1996  4692              F10HC5: ;10-HI-DEN-COMPAT SKIP COUNT =5
1997  4692  40                  .BYTE   @100    ;DATA IS
1998  4693  00                  .BYTE   0       ;       F,F,F,F,F,1,2,3,4
1999  4694  00                  .BYTE   0       ;ONLY GET LAST 2 SCLKS
2000  4695  12                  .BYTE   @22     ;2ND SCLK
2001  4696  0C                  .BYTE   @14
2002  4697  00                  .BYTE   0
2003
2004  4698              F10HD6:
2005  4698              F10HC6: ;10-HI-DEN-COMPAT SKIP COUNT -6
2006  4698  00                  .BYTE   0       ;DATA IS
2007  4699  00                  .BYTE   0       ;       F,F,F,F,F,F,1,2,3
2008  469A  00                  .BYTE   0       ;ONLY GET LAST 2 SCLKS
2009  469B  0D                  .BYTE   @15     ;2ND SCLK
2010  469C  08                  .BYTE   @10
2011  469D  00                  .BYTE   0
2012
```

```
2013  469E                          F10HD7:
2014  469E                          F10HC7: ;10-HI-DEN-COMPAT SKIP COUNT =7
2015  469E  00                              .BYTE   0       ;DATA IS
2016  469F  00                              .BYTE   0       ;       F,F,F,F,F,F,F,1,2
2017  46A0  00                              .BYTE   0       ;ONLY GET LAST 2 SCLKS
2018  46A1  08                              .BYTE   @10     ;2ND SCLK
2019  46A2  04                              .BYTE   @4
2020  46A3  00                              .BYTE   0
2021
2022  46A4                          F10HD8:
2023  46A4                          F10HC8: ;10-HI-DEN-COMPAT SKIP COUNT =8
2024  46A4  00                              .BYTE   0       ;DATA IS
2025  46A5  00                              .BYTE   0       ;       F,F,F,F,F,F,F,F,1
2026  46A6  00                              .BYTE   0       ;ONLY GET LAST 2 SCLKS
2027  46A7  04                              .BYTE   @4      ;2ND SCLK
2028  46A8  00                              .BYTE   0
2029  46A9  00                              .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*****************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*****************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*****************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*****************************************************************************
                                12.0    CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                                12.0            PUSH    B
                                 7.0            MVI     B,0             ;START TO CLEAR AT PORT #0
                                10.0    CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
                                 7.0            ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                                 4.0            ORA     B               ;ADD IN THE SELECTED PORT #
                                10.0            OUT     MBSEL           ;RESET TO THIS PORT
                                 7.0            MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                                10.0            OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                                 4.0            XRA     A               ;CLEAR TU COMMAND A
                                10.0            OUT     TCMD            ;
                                 7.0            MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                                10.0            OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                                 7.0            MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
                                10.0            OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
                                 4.0            XRA     A
                                10.0            OUT     TAMT            ;CLEAR AMTIE WORD
                                10.0            OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                                10.0            OUT     PENAB           ;CLEAR PORT ENABLE WORD
                                 4.0            INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                                 4.0            MOV     A,B
                                 7.0            CPI     4               ;DONE?
                                10.0            JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                                10.0            POP     B               ;RESET B & C
                                10.0            POP     PSW             ;ALL DONE
                                10.0            RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
46D5                            SSUB
                                ;*****************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
46D5                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
46D5                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 9
                                ;*    BGNDO
                                ;*     :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*     :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*     :    GET THE LEAST SIGNIFICANT BIT OF FACH (INPUT) DATA BYTE AND
                                ;*     :        FORM A TABLE INDEX (MS BIT OF INDEX - LS BIT OF THE FIRST
                                ;*     :        BYTE OF THE INPUT DATA, ETC.)
                                ;*     :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*     :        TO BUILD THE INDEX)
                                ;*     :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*     :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*     :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*     :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*     :        DATA BYTE.
                                ;*     :    DECREMENT THE LOOP COUNT
                                ;*     :    DO UNTIL THE LOOP COUNT - 0
                                ;*    ENDDO
                                ;*FNDSUB
46D5                            S
                                ;*****************************************************************************

46D5  F5              12.0  T4X5:   PUSH    PSW             ;SAVE REGISTER A + PSW
46D6  C5              12.0          PUSH    B               ;SAVE REGISTER B + C
46D7  D5              12.0          PUSH    D               ;SAVE REGISTER D + E
46D8  E5              12.0          PUSH    H               ;SAVE REGISTER H + L
                                ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                                ;CLEAR THE TRANSLATED DATA TABLE.
46D9  06 0A            7.0          MVI     B,10            ;SET UP LOOP COUNT
46DB  11 65 47        10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
46DE  AF               4.0          XRA     A               ;CLEAR A
46DF  12               7.0  D4X5:   STAX    D               ;STORE IN THE TABLE
46E0  13               6.0          INX     D               ;UPDATE TABLE POINTER
46E1  05               4.0          DCR     B               ;DECREMENT LOOP COUNT
46E2  C2 DF 46        10.0          JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
46E5  0E 09            7.0          MVI     C,9             ;SET UP TRACK COUNT
```

```
2119  46E7  06  04          7.0  B4X5:  MVI   B,4          ;SET UP BIT COUNT
2120  46E9  11  61  47      10.0         LXI   D,TRNIN
2121  46EC  AF               4.0         XRA   A            ;CLEAR THE GROUP POSITION COUNT
2122  46ED  32  6F  47      13.0         STA   GP4X5
2123  46F0  1A               7.0  C4X5:  LDAX  D            ;GET A DATA BYTE
2124  46F1  A7               4.0         ANA   A            ;SET CONDITION BITS
2125  46F2  E2  F6  46      10.0         JPO   PO4X5        ;ODD PARITY LEAVE CARRY CLEAR
2126  46F5  37               4.0         STC                ;EVEN PARITY SET CARRY
2127  46F6  1F               4.0  PO4X5: RAR                ;SHIFT OUT DESIRED BIT
2128  46F7  12               7.0         STAX  D            ;STORE RESULT BACK IN TEMP TABLE
2129  46F8  3A  6F  47      13.0         LDA   GP4X5        ;GET THE GROUP POSITION BYTE
2130  46FB  17               4.0         RAL                ;PUT IN THIS DATA BIT
2131  46FC  32  6F  47      13.0         STA   GP4X5        ;SAVE THE UPDATED GROUP POSITION BYTE
2132  46FF  13               6.0         INX   D            ;UPDATE THE TABLE POINTER
2133  4700  05               4.0         DCR   B            ;DECREMENT THE BIT COUNT
2134  4701  C2  F0  46      10.0         JNZ   C4X5         ;DO UNTIL ALL 4 BITS ARE RECEIVED
2135
2136                                    ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2137
2138                                    ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2139  4704  06  05          7.0         MVI   B,5          ;SET UP LOOP COUNT
2140  4706  11  65  47      10.0         LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2141  4709  13               6.0  E4X5:  INX   D            ;POINT TO PARITY BIT
2142  470A  1A               7.0         LDAX  D            ;GET PARITY BIT
2143  470B  A7               4.0         ANA   A            ;CLEAR THE CARRY BIT
2144  470C  1F               4.0         RAR                ;SHIFT TO THE CARRY BIT
2145  470D  12               7.0         STAX  D            ;STORE IT BACK
2146  470E  1B               6.0         DCX   D            ;DECREMENT TO DATA BITS
2147  470F  1A               7.0         LDAX  D            ;GET THE DATA BITS
2148  4710  1F               4.0         RAR                ;SHIFT IN CARRY BITS
2149  4711  12               7.0         STAX  D            ;STORE BACK IN TABLE
2150  4712  13               6.0         INX   D            ;POINT TO NEXT TABLE ENTRY
2151  4713  13               6.0         INX   D            ;
2152  4714  05               4.0         DCR   B            ;DECREMENT LOOP COUNT
2153  4715  C2  09  47      10.0         JNZ   E4X5         ;DO UNTIL LOOP COUNT=0
2154  4718  11  45  47      10.0         LXI   D,TAB4X5     ;LOAD ADDRESS OF TRANSLATION TABLE
2155  471B  26  00           7.0         MVI   H,0          ;CLEAR REGISTER H
2156  471D  3A  6F  47      13.0         LDA   GP4X5        ;GET GROUP POSITION COUNT
2157  4720  6F               4.0         MOV   L,A
2158  4721  19              10.0         DAD   D            ;ADD GROUP POSITION COUNT TO TABLE
2159  4722  7E               7.0         MOV   A,M          ;GET TRANSLATED DATA
2160  4723  32  70  47      13.0         STA   TRNTMP
2161                                    ;''TRNTMP'' = THE TRANSLATED DATA
2162
2163  4726  11  65  47      10.0         LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2164  4729  06  05           7.0         MVI   B,5          ;SET UP LOOP COUNT
2165  472B  3A  70  47      13.0  F4X5:  LDA   TRNTMP       ;GET TRANSLATED DATA
2166  472E  13               6.0         INX   D            ;POINT TO PARITY POSITION
2167  472F  A7               4.0         ANA   A            ;CLEAR CARRY BIT
2168  4730  1F               4.0         RAR                ;SHIFT OUT A BIT
2169  4731  32  70  47      13.0         STA   TRNTMP       ;STORE TRANSLATED DATA
2170  4734  1A               7.0         LDAX  D            ;GET THE PARITY BIT (ALWAYS ZERO)
2171  4735  17               4.0         RAL                ;ROLL IN THE CARRY BIT
2172  4736  12               7.0         STAX  D            ;STORE AS THE PARITY BIT
```

```
2173  4737  13          6.0   INX   D      ;UPDATE TABLE POINTER
2174  4738  05          4.0   DCR   B      ;DECREMENT THE LOOP COUNT
2175  4739  C2  2B  47  10.0   JNZ   F4X5   ;DO UNTIL LOOP COUNT=0
2176
2177              ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2178
2179  473C  0D          4.0   DCR   C      ;DECREMENT THE TRACK COUNT
2180  473D  C2  E7  46  10.0   JNZ   B4X5   ;DO UNTIL ALL TRACKS TRANSLATED
2181  4740  E1         10.0   POP   H      ;RESTORE REGISTER H + L
2182  4741  D1         10.0   POP   D      ;RESTORE REGISTER D + E
2183  4742  C1         10.0   POP   B      ;RESTORE REGISTER B + C
2184  4743  F1         10.0   POP   PSW    ;RESTORE REGISTER A + PSW
2185  4744  C9         10.0   RET          ;RETURN TO USER
```

```
                                .SBTTL  TABLE 4 X 5 TRANSLATION

                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
                                ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
                                ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
                                ;
                                ;       0000                              10011
                                ;       0001                              11011
                                ;       0010                              01001
                                ;       0011                              11001
                                ;       0100                              10111
                                ;       0101                              10101
                                ;       0110                              01101
                                ;       0111                              11101
                                ;       1000                              01011
                                ;       1001                              10010
                                ;       1010                              01010
                                ;       1011                              11010
                                ;       1100                              01111
                                ;       1101                              10110
                                ;       1110                              01110
                                ;       1111                              11110

4745  13                        TAB4X5: .BYTE   @23
4746  18                                .BYTE   @33
4747  09                                .BYTE   @11
4748  19                                .BYTE   @31
4749  17                                .BYTE   @27
474A  15                                .BYTE   @25
474B  0D                                .BYTE   @15
474C  1D                                .BYTE   @35
474D  0B                                .BYTE   @13
474E  12                                .BYTE   @22
474F  0A                                .BYTE   @12
4750  1A                                .BYTE   @32
4751  0F                                .BYTE   @17
4752  16                                .BYTE   @26
4753  0E                                .BYTE   @16
4754  1E                                .BYTE   @36
```

```
                                        .SBTTL  SUBROUTINE VARIABLES

                                SDDRA:  .BYTE   0               ;STORAGE FOR DDR A DATA
                                SDDRB:  .BYTE   0               ;STORAGE FOR DDR B DATA
                                SDDRC:  .BYTE   0               ;STORAGE FOR DDR C DATA
                                SCLKCT: .BYTE   0               ;COUNT OF 'SCLK' RECEIVED
                                ESCLK:  .BYTE   0               ;NUMBER OF EXPECTED SCLK PER PASS
                                FMTNUM: .BYTE   0               ;SELECTED PACKING MODE PER PASS
                                SKIPCT: .BYTE   0               ;SKIP COUNT IN USE FOR CURRENT XFR
                                EXPADR: .ADDR   0               ;ADDRESS OF THE EXPECTED DATA TABLE
                                UNITMP: .BYTE   0               ;PORT NUMBER TO USE AS RESPONDED BY HOST
                                SUNIT:  .BYTE   0               ;STORAGE OF THE SELECTED UNIT NUMBER
                                CCTLWD: .BYTE   0               ;LOAD INTO THE SYSTEM CLOCK CONTROL WORD

                                TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
                                                                ;BE TRANSLATED

                                TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
                                                                ;TRANSLATION

                                GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
                                                                ;BE TRANSLATED

                                TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
```

```
2249                                    .SBTTL  SUBROUTINE CLEAR ECC
2250  4771                              S
 (1)                                    ; ************************************************************************
2251                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION 'ECCCHR' -
2252                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2253                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2254  4771                              S
 (1)                                    ; ************************************************************************
2255
2256  4771  AF                   4.0  CLECC:  XRA     A              ;CLEAR THE ACCUMULATOR
2257  4772  32  9D  47            13.0         STA     ECCCHR         ;CLEAR THE ECC CHARACTER
2258  4775  C9                    10.0         RET                    ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        S
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF 'ECCCHR'' TO UPDATE THE CONTENTS OF 'ECCCHR'' ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        S
                                        ; ************************************************************************
                                ECC:    PUSH    H
                                        PUSH    B
                                        LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
                                        XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
                                        MOV     E,A             ;SAVE XOR RESULT IN E
                                        ANI     $10             ;IS BIT #4 OF RESULT SET
                                        MOV     A,E             ;RESTORE XOR RESULT FROM B
                                        JZ      ECC1            ;CONTINUE IF BIT #4 RESET
                                        XRI     $23             ;ELSE-XOR WITH 23
                                ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                        XRA     A               ;CLEAR A
                                        MOV     C,A             ;CLEAR THE TRANSLATE RESULT
                                        LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
                                        CALL    TRANS           ;TRANSLATE THE BITS
                                        MOV     A,C             ;GET THE TRANSLATED RESULT
                                        STA     ECCCHR          ;STORE RESULT
                                        POP     B
                                        POP     H
                                        RET

                                ECCTBL: $08                     ;BIT 0 = POSITION 3
                                        $20                     ;BIT 1 = POSITION 5
                                        $02                     ;BIT 2 = POSITION 1
                                        $40                     ;BIT 3 = POSITION 6
                                        $80                     ;BIT 4 = POSITION 7
                                        $01                     ;BIT 5 = POSITION 0
                                        $10                     ;BIT 6 - POSITION 4
                                        $04                     ;BIT 7 = POSITION 2

                                ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

479E  06 01           7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
47A0  7B              4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
47A1  A0              4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
47A2  CA A8 47       10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
47A5  79              4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
47A6  B6              7.0          ORA   M            ;'OR' IN NEW POSITION
47A7  4F              4.0          MOV   C,A          ;SAVE RESULT

47A8  78              4.0  TRANS2: MOV   A,B
47A9  07              4.0          RLC                ;POSITION MASK TO NEXT BIT
47AA  47              4.0          MOV   B,A
47AB  D8             12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
47AC  23              6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
47AD  C3 A0 47       10.0          JMP   TRANS1       ;PROCESS NEXT BIT

47B0                                S
 (1)                                ; ***************************************************************************
                                    ;CLKSYS - ROUTINE TO CLOCK THE SYSTEM USING THE CONTENTS OF 'CCTLWD'
                                    ;       AS A BASE AND TOGGLING THE CLOCK BIT IN THE CLOCK CONTROL WORD
47B0                                S
 (1)                                ; ***************************************************************************

47B0  F5             12.0  CLKSYS: PUSH  PSW
47B1  3A 60 47       13.0          LDA   CCTLWD       ;GET THE CONTROL WORD BASE
47B4  F6 40           7.0          ORI   SSCLK        ;GET THE CLOCK BIT
47B6  D3 F0          10.0          OUT   CLKCTL       ;SET THE CLOCK ON
47B8  E6 BF           7.0          ANI   @277         ;CLEAR THE CLOCK BIT
47BA  D3 F0          10.0          OUT   CLKCTL       ;FINISH THE CLOCK CYCLE
47BC  F1             10.0          POP   PSW
47BD  C9             10.0          RET
      0000                  .END
```

```
A      =%0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
ARAIDF= 0098    ASAVE   4F9B    ATTCD   4F97    AXNUM   4F91
B      =%0000   BADST  = 0090   BIT0   = 0001   BIT1   = 0002
BIT15 = 8000    BIT2   = 0004   BIT3   = 0008   BIT4   = 0010
BIT5   = 0020   BIT6   = 0040   BIT7   = 0080   BIT8   = 0100
BIT9   = 0200   BRKPBC= 4F0A    BRKRAM= 4F10    BRKSTR= 4E60
BRKXCT= 4F00    BSAVE   4F9C    BYTCNT= 00D4    BYTEH   4F24
BYTEL   4F23    B4X5    46E7    C      =%0001   CASCT   4F21
CASCTL= 00A0    CASSTA= 00A0    CATTH  = 0089   CATTL  = 0088
CBUSST= 00A1    CBYTH  = 008B   CBYTL  = 008A   CCTLWD  4760
CDG1H  = 0087   CDG1L  = 0086   CDG2H  = 0093   CDG2L  = 0092
CDG3H  = 0095   CDG3L  = 0094   CDVTH  = 008D   CDVTL  = 008C
CHPTIE= 0028    CHOTIE= 0020    CH1TIE= 0021    CH2TIE= 0022
CH3TIE= 0023    CH4TIE= 0024    CH5TIE= 0025    CH6TIE= 0026
CH7TIE= 0027    CKLOP  = 2815   CLEAR   46AA    CLECC   4771
CLKCTL= 00F0    CLKRS   43E5    CLKSYS  47B0    CLOCK   4F26
CLRLP   46AE    CMC0H  = 0099   CMC0L  = 0098   CMC1H  = 009B
CMC1L  = 009A   CMC2H  = 009D   CMC2L  = 009C   CMC3H  = 009F
CMC3L  = 009E   CMINH  = 0097   CMINL  = 0096   CNTCTL= 00D7
CNV1    456F    CCNVRT  4569    CRCWRD= 0018    CSAVE   4F9D
CSRLH  = 0091   CSRLL  = 0090   CTCH   = 0085   CTCL   = 0084
CTSTH  = 008F   CTSTL  = 008E   CXCTH  = 0081   CXCTL  = 0080
CXINH  = 0083   CXINL  = 0082   C.     = 0001   C.AVAI= 0080
C.DP   = 0008   C.DSE  = 0010   C.DTU  = 0003   C.DVA  = 0008
C.FAIL= 00FC    C.FMT  = 0070   C.FNCT= 003E   C.GO   = 0001
C.INTC= 00FE    C.MAIN= 0020    C.NSA  = 0080   C.RCT  = 00FC
C.SER  = 0080   C.SHR  = 0040   C.SKPC= 000F   C.TAPE= 0040
C.WCS  = 0002   C1EC1   43F0    C1EC1L  43F8    C1EC2   4407
C1EC2L  440F    C4X5    46F0    D      =%0002   DATACT= 00D0
DBUS    4F28    DBUSCT= 00C0    DBUSST= 00C0    DDRA   = 00D8
DDRAIN= 0010    DDRB   = 00D9   DDRBIN= 0002    DDRC   = 00DA
DDRCIN= 0001    DDRCO  = 0088   DDRCTL= 00DB    DG1     4595
DG1ECC  459C    DG2     459D    DG2ECC  45A4    DIAFLG  4F22
DIAGPG= 4300    DIAGRM= 4F90    DIARD  = 0008   DONE1  = 0045
DONINT= 0010    DSAVE   4F9E    DSE    = 0006   DUMM1   447D  G
DUMM2   449D  G  D.ATH0= 0001   D.ATH1= 0002   D.EOTD= 0010
D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004   D.TACH= 0008
D.WR4  = 0080   D4X5    46DF    E      =%0003   ECC     4776
ECCBAD= 0042    ECCCHR  479D    ECCCOR= 0019    ECCOK  = 0041
ECCSTA= 001A    ECCTBL  4795    ECCTST= 000E    ECC1    4785
EDATA  = 0095   EOTCLR= 0003    ERFLG   4F93    ERLP   = 2809
ERLPA  = 280F   ERLPB  = 2812   ERLPE  = 280C   ERNUM   4F90
ERRCNT= 00D6    ESAVE   4F9F    ESCLK   4759    EXIT    454F
EXPADR  475C    E.ACRC= 0010    E.AMT  = 0020   E.CDP  = 0080
E.CRC  = 0080   E.PNTR= 0008    E.RPE  = 0040   E.STEC= 0001
E.TTEC= 0002    E.UNC  = 0004   E4X5    4709    FIFORD= 006A
FMTNUM  475A    FORMAT  4F25    FOUND   4327    FPORT   4347
FWDTST= 0061    F10C1   4632    F10C2   4638    F10C3   463E
F10D1   4644    F10D2   464A    F10D3   4650    F10D4   4656
F10HC1  465C    F10HC2  4668    F10HC3  4674    F10HC4  4680
F10HC5  4692    F10HC6  4698    F10HC7  469E    F10HC8  46A4
F10HD1  465C    F10HD2  4668    F10HD3  4674    F10HD4  468C
F10HD5  4692    F10HD6  4698    F10HD7  469E    F10HD8  46A4
F11N    4626    F15N    462C    F4X5    472B    GCRID  = 0089
```

```
GCRSET= 0002
HLSAVE  4FA0
I.PWR = 0020
I7.5  = 0040
KENAB = 0078
KEY10 = 006D
KEY14 = 005D
KEY18 = 003D
KEY3  = 007A
KEY7  = 0076
KLDAD = 003D
KN3   = 005E
KN7   = 0074
KU3   = 007A
LCE   = 000B
LC0   = 0000
LC4   = 0004
LC8   = 0008
LDLEDB= 00CB
LDLEDF= 00CF
LKLWMG= 0058
LKMOD7= 0046
LPNUM   4F92
MB.B  - 0004
MSE   - 0008
MT.CPE- 0080
MT.LWR= 0004
MT.PSB= 0004
MT.WRT- 0010
M.CONT= 0080
M.FAIL- 0008
M.ONLI= 0001
M.RDPE= 0008
M.UNIT= 0007
M5.5  = 0001
NOUNIT  43'E  G
OPVER = 0040
PEID  - 008A
PO4X5   46F6
PSTAT = 0048
P.BCTC- 0040
P.LWR - 0020
P.RPOE- 0020
P.SING= 0080
P.TUPR- 0010
P.WPEN= 0010
P.WP3E- 0004
RAMT  - 0010
RCHBD1- 0047
RCMD  = 0008
RDCLK = 0010
REQST = 2806
RFIFOL- 0008
RILL  - 0012

GOODTM= 0092
IE    = 0008
I.RMPE= 0040
KCALL = 005F
KEXAM = 003E
KEY11 = 006E
KEY 5 = 005E
KEY19 = 003E
KEY4  = 007B
KEY8  = 0077
KN0   = 003C
KN4   - 006C
KN8   = 0075
KU8   = 0077
LCH   = 000C
LC1   = 0001
LC5   = 0005
LC9   = 0009
LDLEDC= 00CC
LKDIAG= 2800
LKLWMP= 0055
LKOPR = 0046
M     =%0006
MEMTOP= 4FFF
MSGN  = 0006
MT.DSE= 0001
MT.MOT= 0002
MT.PS0= 0001
MT.Z  = 0008
M.DEM = 0020
M.ILR = 0010
M.PE  = 0040
M.RUN - 0004
M.WCLK= 0040
M6.5  - 0002
OKAY  = 00FF
PADCNT- 00D5
PENAB = 004C
PRDD  - 004C
PSW   -%0009
P.CMDP= 0020
P.RDP = 0002
P.RP1E= 0010
P.STAT= 0002
P.WCSP= 0004
P.WP0E= 0008
P.5VOK= 0002
RARA  - 0006
RCHOK = 0046
RCMLP - 0003
RDON  = 0011
RESCHR  00D1
RGCLK = 0002
RINST = 000C

GP4X5   476F
INTSTA- 00E0
I5.5  = 0010
KCLR  = 007B
KEYBRD= 00C8
KEY12 = 006F
KEY16 = 005F
KEY2  = 0079
KEY5  = 0074
KEY9  = 006C
KN1   = 005C
KN5   = 006D
KN9   = 0076
L     =%0005
LCL   = 000D
LC2   = 0002
LC6   = 0006
LDFIFO  4583
LDLEDD= 00CD
LKKBD = 004C
LKLWPG= 0052
LOADIT  457E
MBSEL = 00E0
MINUS = 000A
MTACLR= 0000
MT.FWD= 0040
MT.NWT= 0080
MT.PS1= 0002
M.ATA = 0080
M.EBL = 0010
M.INIT= 0010
M.PORT= 0080
M.SCLK= 0001
M.WCLN= 0080
M7.5  = 0004
OPRRAM= 4300
PADCRC= 0080
PESET = 0001
PRENF = 009C
PTABLE  4624
P.INTE= 0080
P.RPEN- 0004
P.RP2E= 0020
P.STPE= 0080
P.WDS = 0040
P.WP1E= 0004
QUE   = 281B
RARAI = 0004
RCHTST= 000C
RCONT = 0080
READG = 0007
REVTST= 0064
RGCRI = 0003
RMCTST= 0008

H     =%0004
ITERA   4F9A
I6.5  = 0020
KDEP  = 003F
KEY1  = 0078
KEY13 = 005C
KEY17 = 003C
KEY20 = 003F
KEY6  = 0075
KINTA = 006F
KN2   = 005D
KN6   = 006E
KU2   = 0079
LBLANK= 000F
LCP   = 000E
LC3   = 0003
LC7   = 0007
LDLEDA= 00CA
LDLEDE= 00CE
LKKEY = 0049
LKLWPP= 004F
LPFLG   4F94
MB.A  = 0008
MM    = 8000
MT.ARA= 0020
MT.INH= 0008
MT.PEC= 0040
MT.REV= 0020
M.CAPE= 0020
M.EXC = 0008
M.OCC = 0020
M.RDEN= 0002
M.TRA = 0040
M.WREN= 0080
NOTCAP= 0088
OPSTRT= 0058
PDIAG = 0048
PL    = 00B1
PS    = 00B2
P.AMTP= 0001
P.LCS = 0040
P.RPST= 0002
P.RP3E= 0010
P.TACH= 0008
P.WFLP= 0001
P.WP2E= 0008
QUEM  = 281E
RCHBD0= 0048
RCLRT = 000D
RDATA = 0017
REND  = 0014
REWIND= 0004
RIBG  = 0001
RMK2  - 0013
```

```
RNOP   = 0000    RPATH  = 0001    RPBAD  = 0044    RPCHI  = 0001
RPCLK  = 0003    RPCTL  = 0009    RPEI   = 0002    RPFAIL= 0000
RPF1   = 009D    RPF2   = 009E    RPOK   = 0043    RPOSTN= 0016
RPSTA  = 0015    RRCMT  = 000A    RSTAT  = 0002    RTIEB  = 000A
RTIER  = 0030    RTM    = 0005    RUNKI  = 0009    RUPTST= 005E
RWDUNL= 0005     R.AMT  = 0001    R.BOP  = 0008    R.DATA= 0040
R.DON  = 0002    R.DRDY= 0010     R.END  = 0010    R.ILL  = 0004
R.JVOK= 0004     R.MK2  = 0008    R.PLOD- 0008     R.PLOO= 0010
R.PLO1= 0020     R.POST= 0020     R.STNM= 0002     R.STOP= 0004
R.STPC= 0001     R.TBJN= 0080     R.TSTD= 0040     R.VOK  = 0080
R00H   = 0081    R00L   = 0080    R01H   = 0083    R01L   = 0082
R02H   = 0085    R02L   = 0084    R03H   = 0087    R03L   = 0086
R04H   = 0089    R04L   = 0088    R05H   = 008B    R05L   = 008A
R06H   = 008D    R06L   = 008C    R07H   = 008F    R07L   = 008E
R10H   = 0091    R10L   = 0090    R11H   = 0093    R11L   = 0092
R12H   = 0095    R12L   = 0094    R13H   = 0097    R13L   = 0096
R14H   = 0099    R14L   = 0098    R15H   = 009B    R15L   = 009A
R16H   = 009D    R16L   = 009C    R17H   = 009F    R17L   = 009E
R7.5   = 0010    SCLKCT  4758     SDDRA   4755     SDDRB   4756
SDDRC   4757     SELCLR= 0000     SETATA- 00A1     SG2     4599
SG4     45A1     SID    = 0080    SKIPCT  475B     SOD    = 0080
SOE    = 0040    SP     =%0008    SSCLK  = 0040    SSTEP  = 0005
STACK  - 4FFF    STATRM= 4F20     STPCT   4F20     STRSP  = 5000
SUNIT   475F     TAB4X5  4745     TADR00= 0080     TADR01= 0081
TADR02  0082     TADR03= 0083     TADR04= 0084     TADR05= 0085
TADR06= 0086     TADR07= 0087     TADR10= 0088     TADR11= 0089
TADR12= 008A     TADR13= 008B     TAMT   - 0044    TASEL  = 0080
TCMD   = 0040    TC.FWD= 0040     TC.INH= 0008     TC.LWR= 0004
TC.REV= 0020     TC.WRT= 0010     TEMP    4F99     TEST1   4300
TMF    = 0099    TMRDY  = 0040    TRANS   479E     TRANS1  47A0
TRANS2  47A8     TRKENA= 00D2     TRNIN   4761     TRNOUT  4765
TRNTMP  4770     TSET   = 2803    TSTEND= 2818     TSTS   = 0040
TUSEL0= 00D1     TUSEL1= 00D2     TU78   = 0010    T.ATH0= 0001
T.ATH1= 0002     T.BOT  = 0004    T.EOT  = 0002    T.FPT  = 0007
T.NTHR- 0004     T.ONL  - 0020    T.PES  = 0008    T.PSBJ= 0020
T.PSOJ- 0008     T.PS1J= 0010     T.RDY  = 0080    T.RDYO= 0040
T.RWD  = 0010    T.SCLK= 0002     T1CKA   44D0     T1CKAC  44F9  G
T1CKB   44FF     T1CKBC  4513  G  T1CKC   4519     T1CKCC  452D  G
T1CLK   4468     T1CLKA  4462     T1LOOP  436D     T1PLO   441E
T1SCKL  4488     T1SCK0  44A6     T1SCLK  4486     T1SG1   4436
T1SG2   443F     T1SG3   4448     T1SG4   4451     T1ST    43EE
T4X5    46D5     UIBG   = 00A1    UNITMP  475E     VALFC   4F98
VALTB   4F95     VELTST= 005B     WDR.P  = 0010    WMCCTL= 00D3
WMCERR- 00DA     WMCSTA= 00D0     WRTCLK= 0000     WRTDAT= 00D3
W.ACRC= 0004     W.CRC  = 0008    W.DIAG= 0002     W.DONN= 0040
W.ECC   0010     W.ENAB= 0080     W.ERR  = 0020    W.FMT  = 0070
W.GCR  - 0010    W.LEFT= 0004     W.ONES= 0020     W.RESI= 0002
W.REV  = 0004    W.ROME= 0010     W.RST  = 0001    W.SKIP= 000F
W.WRIT- 0008     W.XFER= 0020     X      -%000A    XFRPTB  45A5
X.DONN- 0080     X.ENAB- 0040     X.PEPE- 0002     X.ROME- 0001
X.WCLK- 0001     Y      %000B     .      - 47BE
```

```
 ERRORS DETECTED:  0

 *WMC8.A78/PTP,WMC8=NLIST,PARAM,MACRO,LIST,WMC8
 RUN-TIME: 4 5 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:01 SEQ 1190
PARAM.M80 TABLE OF CONTENTS

```
                                .RADIX 16
                                .TITLE  RPM6 - READ PATH MICRO DIAGNOSTIC PART #6
                                ;ID     RPM6-READ PATH MICRO DIAGNOSTIC PART #6

                                ;DEFINITIONS

                                        I01=@001                        ;INTERUPT CODE VALUES
0001                                    I25=@25
0015                                    I30-@30
0018

                                        F03=@3*400                      ;FAILURE CODE EQUIVALENTS
0C00                                    F04=@4*400
1000                                    F05=@5*400
1400                                    F06=@6*400
1800                                    F07=@7*400
1C00                                    F10=@10*400
2000                                    F11=@11*400
2400                                    F12-@12*400
2800                                    F21=@21*400
4400                                    F22=@22*400
4800                                    F23=@23*400
4C00

                                ;SUBROUTINE LINK TABLE ADDRESSES

                                        WRTSET  =       080C
080C
```

```
1356                                    .SBTTL  TEST 1 - PHASE LOCKED LOOP TEST PART 1
1357  4300                              ST
 (1)                                    ;*****************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1358                                    ;*PHASE LOCKED LOOP TEST PART 1
1359  4300                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1360                                    ;*THIS TEST PERFORMS A GCR LOOP WRITE TO READ
1361                                    ;*WITH PLO BYPASS 01 SET
1362                                    ;*IF NO FAILURE OCCURS THEN TEST 2 WILL BE
1363                                    ;*PERFORMED TO RUN A LWR WITH THE PLL'S ENABLED
1364  4300                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1365                                    ;*BGNTST
1366                                    ;*  SET CLOCKS TO NORMAL
1367                                    ;*  SET THE BYTE COUNT TO 55.
1368                                    ;*  CLEAR ALL AMTIE TEMP LOCATIONS
1369                                    ;*  CLEAR ALL RAM TEMP LOCATIONS
1370                                    ;*  SET XFRCTL SO PLL'S ARE DISABLED
1371                                    ;*  ISSUE A LOOP WRITE TO READ
1372                                    ;*  IF ANY ERRORS OCCURED
1373                                    ;*  :  THEN
1374                                    ;*  :    REPORT THE ERROR
1375                                    ;*  :  ELSE
1376                                    ;*  :    CONTINUE
1377                                    ;*  ENDIF
1378                                    ;*ENDTST
1379  4300                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1380                                    ;*RPM6 MICRO TEST 01
1381                                    ;*RPM6 MICRO ERROR 01
1382                                    ;*RPM6-PHASE LOCKED LOOPS TEST PART 1
1383                                    ;*M8950'S , M8953
1384                                    ;*OPERATOR ERROR NO TU78 UNITS SPECIFIED
1385                                    ;*FATAL ERROR - TEST ABORTED
1386                                    ;*
1387                                    ;*RPM6 MICRO TEST 01
1388                                    ;*RPM6 MICRO ERROR 02
1389                                    ;*RPM6-PHASE LOCKED LOOPS TEST PART 1
1390                                    ;*M8950'S , M8953
1391                                    ;*LOOP WRITE/READ ERROR WITH PLL'S DISABLED
1392                                    ;*ACTUAL = AAAA
1393                                    ;*EXPECTED = EEEE
1394                                    ;*BYTE-SCLK COUNT - LLL
1395                                    ;*DATA FORMAT - MM
1396                                    ;*SKIP COUNT = NN
1397                                    ;*ACTUAL - FAILURE CODE
```

```
1398                                    ;*EXPECTED = INTERUPT CODE
1399                                    ;*
1400                                    ;*RPM6 MICRO TEST 02
1401                                    ;*RPM6 MICRO ERROR 03-11
1402                                    ;*RPM6-PHASE LOCKED LOOPS TEST PART 1
1403                                    ;*M8950'S , M8953
1404                                    ;*LOOP WRITE / READ ERROR WITH PLL DISABLED
1405                                    ;*ACTUAL = AAAA
1406                                    ;*EXPECTED = EEEE
1407                                    ;*BYTE-SCLK COUNT - LLL
1408                                    ;*DATA FORMAT = MM
1409                                    ;*SKIP COUNT = NN
1410                                    ;*CHANNEL (N) FAILED
1411                                    ;*ACTUAL DATA IS CHANNEL (N) TIE BUS REGISTER
1412  4300                              $
 (1)                                    ; ***************************************************************************************
```

```
1414  4300                                  TEST1:  TESTX   @1                      ;INITIALIZE THE TEST
 (1)  4300  3E 01              7.0                     MVI     A,@1                ;DEFINE THE TEST NUMBER
 (1)  4302  CD 03 28          18.0                     CALL    TSET                ;SETUP THE TEST
1415                                        ;%RPM6-PHASE LOCKED LOOPS TEST PART 1
1416                                        ;&M8950'S , M8953
1417  4305                                          REQ     @7,0,0,0,0
 (1)  4305  CD 06 28          18.0                     CALL    REQST
 (1)  4308  00                                         .BYTE   0                   ;DATA PATTERN NUMBER
 (1)  4309  00 00                                      .WORD   0                   ;SYSTEM "0" COUNT
 (1)  4308  00 00                                      .WORD   0                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                         .BYTE   0                   ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                         .BYTE   @7                  ;REQUEST CODE
1418  430F                                          RIN     R12L
 (1)  430F  DB 94              10.0                    IN      R12L                ;READ R12L INTO AC
 (1)  4311  7F                  4.0                    MOV     A,A                 ;RETRY LINK
1419  4312  32 00 49           13.0              STA     UNITMP
1420  4315  A7                  4.0              ANA     A
1421  4316  C2 24 43           10.0              JNZ     TST01
1422  4319                                      ERR     EXIT,DUMMY
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD 09 28          18.0                     CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0001                                       MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                         .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  431D  00                                         .BYTE
 (1)  431E  CD 15 28          18.0             DUMMY:: CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA 31 47          10.0                     JC      EXIT                ;LOOP ADDRESS IF LOOP SPECIFIED
1423                                        ;>OPERATOR ERROR NO TU78 UNITS SPECIFIED
1424                                        ;>FATAL ERROR - TEST ABORTED
1425  4324  21 37 00           10.0  TST01:  LXI     H,55.                       ;SET UP THE BYTE COUNT
1426  4327  22 01 49           16.0              SHLD    SIZE                        ;
1427  432A  3E 01                7.0              MVI     A,W.RST                     ;SET UP WMC RESTART TO STOP
1428  432C  D3 D3               10.0              OUT     WMCCTL                      ;THE WMC WHILE THE CLOCK IS CHANGING
1429  432E  3E 10                7.0              MVI     A,@020                      ;SET UP NORMAL CLOCKS
1430  4330  D3 F0               10.0              OUT     CLKCTL                      ;
1431  4332  AF                   4.0              XRA     A                           ;CLEAR A
1432  4333  D3 D3               10.0              OUT     WMCCTL                      ;CLEAR RESTART BIT
1433  4335  32 0C 49           13.0              STA     SRPFAL                      ;CLEAR TEMP RAM
1434  4338  32 0D 49           13.0              STA     SRPATH
1435  433B  32 FE 48           13.0              STA     TSTAMT
1436  433E  32 FF 48           13.0              STA     TSTAMP
1437  4341  3E FF                7.0              MVI     A,0FF                       ;
1438  4343  32 08 49           13.0              STA     PLLFLG                      ;SET PLL FLAG
1439  4346  32 09 49           13.0              STA     WFFLG                       ;SET FLAG SO WRITE FAIL ERRORS
1440                                                                                ;WILL NOT SHOW UP AS LWR ERROR
1441  4349  3E 18                7.0              MVI     A,R.PLOD+R.PL00             ;GET PLO DISABLE AND PLO BYPASS
1442  434B  32 07 49           13.0              STA     XFRCTL                      ;STORE IN RAM FOR SUBROUTINE
1443  434E  CD 34 47           18.0              CALL    GCRSUB                      ;SETUP FOR LWR
1444  4351  CD EA 47           18.0              CALL    LWR                         ;DO LWR
1445  4354  D2 70 43           10.0              JNC     TST01A                      ;IF NO ERROR GO TO NEXT TEST
1446  4357  78                   4.0              MOV     A,B                         ;GET THE FAILURE CODE
1447  4358  E6 FC                7.0              ANI     $FC                         ;CLEAR OUT JUNK
1448  435A  0F                   4.0              RRC                                 ;POSITION FOR OUTPUT
1449  435B  0F                   4.0              RRC
1450  435C                                        ROUT    ADATA                       ;STORE IN CAS
```

```
  (1)  435C  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  435E  7F               4.0                MOV     A,A             ;RETRY LINK
1451   435F  79               4.0        MOV     A,C                     ;GET THE INTERUPT CODE
1452   4360  E6  3F            7.0        ANI     $3F                     ;CLEAR OUT JUNK
1453   4362                               ROUT    EDATA                   ;WRITE TO CAS
  (1)  4362  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4364  7F               4.0                MOV     A,A             ;RETRY LINK
1454   4365                               ERRB    TEST1,T1CON1,3
  (1)                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
  (1)  4365  CD  12  28      18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        0002                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4368  02                               .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4369  03                               .BYTE   3               ;PRINT ROUTINE NUMBER
  (1)  436A  CD  15  28      18.0        T1CON1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  436D  DA  00  43      10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1455                                   ;>LOOP WRITE/READ ERROR WITH PLL'S DISABLED
1456                                   ;<ACTUAL = FAILURE CODE
1457                                   ;<EXPECTED - INTERUPT CODE
```

```
1459
1460                                        ;
1461                                        ;CHECK TO SEE IF ANY TIE BUS ERRORS OCCURED
1462                                        ;
1463  4370  3E  03          7.0    TST01A: MVI     A,R.STNM+R.STPC         ;STOP READ PATH SO WE CAN
1464                                                                       ;READ REGISTERS 20-27 , 40,47
1465  4372  D3  09         10.0            OUT     RPCTL
1466  4374  AF             4.0            XRA     A                       ;
1467  4375                                 ROUT    EDATA                   ;WRITE EXPECTED DATA
 (1)  4375  D3  95         10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4377  7F             4.0                    MOV     A,A             ;RETRY LINK
1468  4378  DB  20         10.0            IN      CH0TIE                  ;READ CH 0 TIE BUS REG
1469  437A  A7             4.0            ANA     A                       ;ANY ERROR BITS
1470  437B  CA  86  43      10.0            JZ      CH0CON                  ;NO , CONTINUE
1471  437E                                 ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  437E  D3  94         10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4380  7F             4.0                    MOV     A,A             ;RETRY LINK
1472  4381                                 ERRB    TEST1,CH0CON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4381  CD  12  28      18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0003                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4384  03                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4385  03                                            .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4386  CD  15  28      18.0                    CH0CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4389  DA  00  43      10.0                            JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1473                                    ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1474                                    ;<CHANNEL 0 FAILED
1475                                    ;<ACTUAL DATA IS CHANNEL 0 TIE BUS REGISTER
1476  438C  DB  21         10.0            IN      CH1TIE                  ;READ CH 1 TIE BUS REG
1477  438E  A7             4.0            ANA     A                       ;ANY ERROR BITS
1478  438F  CA  9A  43      10.0            JZ      CH1CON                  ;NO , CONTINUE
1479  4392                                 ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4392  D3  94         10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4394  7F             4.0                    MOV     A,A             ;RETRY LINK
1480  4395                                 ERRB    TEST1,CH1CON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4395  CD  12  28      18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4398  04                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4399  03                                            .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  439A  CD  15  28      18.0                    CH1CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  439D  DA  00  43      10.0                            JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1481                                    ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1482                                    ;<CHANNEL 1 FAILED
1483                                    ;<ACTUAL DATA IS CHANNEL 1 TIE BUS REGISTER
```

```
1485  43A0  DB  22          10.0        IN      CH2TIE                  ;READ CH 2 TIE BUS REG
1486  43A2  A7               4.0        ANA     A                       ;ANY ERROR BITS
1487  43A3  CA  AE  43       10.0        JZ      CH2CON                  ;NO , CONTINUE
1488  43A6                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43A6  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43A8  7F               4.0                MOV     A,A             ;RETRY LINK
1489  43A9                               ERRB    TEST1,CH2CON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43A9  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43AC  05                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43AD  03                               .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43AE  CD  15  28       18.0        CH2CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43B1  DA  00  43       10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1490                            ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1491                            ;<CHANNEL 2 FAILED
1492                            ;<ACTUAL DATA IS CHANNEL 2 TIE BUS REGISTER
1493  43B4  DB  23          10.0        IN      CH3TIE                  ;READ CH 3 TIE BUS REG
1494  43B6  A7               4.0        ANA     A                       ;ANY ERROR BITS
1495  43B7  CA  C2  43       10.0        JZ      CH3CON                  ;NO , CONTINUE
1496  43BA                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43BA  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43BC  7F               4.0                MOV     A,A             ;RETRY LINK
1497  43BD                               ERRB    TEST1,CH3CON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43BD  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43C0  06                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43C1  03                               .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43C2  CD  15  28       18.0        CH3CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43C5  DA  00  43       10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1498                            ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1499                            ;<CHANNEL 3 FAILED
1500                            ;<ACTUAL DATA IS CHANNEL 3 TIE BUS REGISTER
```

```
1502  43C8  DB  24        10.0        IN      CH4TIE                  ;READ CH 4 TIE BUS REG
1503  43CA  A7             4.0        ANA     A                       ;ANY ERROR BITS
1504  43CB  CA  D6  43     10.0        JZ      CH4CON                  ;NO , CONTINUE
1505  43CE                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43CE  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43D0  7F             4.0                MOV     A,A             ;RETRY LINK
1506  43D1                             ERRB    TEST1,CH4CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43D1  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43D4  07                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43D5  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43D6  CD  15  28     18.0        CH4CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43D9  DA  00  43     10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1507                             ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1508                             ;<CHANNEL 4 FAILED
1509                             ;<ACTUAL DATA IS CHANNEL 4 TIE BUS REGISTER
1510  43DC  DB  25        10.0        IN      CH5TIE                  ;READ CH 5 TIE BUS REG
1511  43DE  A7             4.0        ANA     A                       ;ANY ERROR BITS
1512  43DF  CA  EA  43     10.0        JZ      CH5CON                  ;NO , CONTINUE
1513  43E2                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43E2  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43E4  7F             4.0                MOV     A,A             ;RETRY LINK
1514  43E5                             ERRB    TEST1,CH5CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43E5  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E8  08                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E9  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43EA  CD  15  28     18.0        CH5CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43ED  DA  00  43     10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1515                             ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1516                             ;<CHANNEL 5 FAILED
1517                             ;<ACTUAL DATA IS CHANNEL 5 TIE BUS REGISTER
```

```
1519  43F0  DB  26        10.0        IN      CH6TIE                  ;READ CH 6 TIE BUS REG
1520  43F2  A7             4.0        ANA     A                       ;ANY ERROR BITS
1521  43F3  CA  FE  43     10.0        JZ      CH6CON                  ;NO , CONTINUE
1522  43F6                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43F6  D3  94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43F8  7F                4.0                MOV     A,A             ;RETRY LINK
1523  43F9                             ERRB    TEST1,CH6CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  43F9  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43FC  09                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43FD  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43FE  CD  15  28     18.0        CH6CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4401  DA  00  43     10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1524                        ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1525                        ;<CHANNEL 6 FAILED
1526                        ;<ACTUAL DATA IS CHANNEL 6 TIE BUS REGISTER
1527  4404  DB  27        10.0        IN      CH7TIE                  ;READ CH 7 TIE BUS REG
1528  4406  A7             4.0        ANA     A                       ;ANY ERROR BITS
1529  4407  CA  12  44     10.0        JZ      CH7CON                  ;NO , CONTINUE
1530  440A                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  440A  D3  94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  440C  7F                4.0                MOV     A,A             ;RETRY LINK
1531  440D                             ERRB    TEST1,CH7CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  440D  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000A                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4410  0A                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4411  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4412  CD  15  28     18.0        CH7CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4415  DA  00  43     10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1532                        ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1533                        ;<CHANNEL 7 FAILED
1534                        ;<ACTUAL DATA IS CHANNEL 7 TIE BUS REGISTER
```

```
1536  4418  DB  28        10.0          IN      CHPTIE                  ;READ PARITY TIE BUS REG
1537  441A  A7             4.0          ANA     A                       ;ANY ERROR BITS
1538  441B  CA  26  44     10.0          JZ      CHPCON                  ;NO , CONTINUE
1539  441E                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  441E  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4420  7F              4.0                  MOV     A,A             ;RETRY LINK
1540  4421                               ERRB    TEST1,CHPCON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4421  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000R                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4424  08                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4425  03                                 .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4426  CD  15  28     18.0          CHPCON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4429  DA  00  43     10.0                  JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1541                                ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1542                                ;<PARITY CHANNEL FAILED
1543                                ;<ACTUAL DATA IS PARITY CHANNEL TIE BUS REGISTER
1544  442C                                  ENDTST  TST01
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  442C                              REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  442C  CD  06  28     18.0                  CALL    REQST
 (2)  442F  00                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  4430  00  00                             .WORD                   ;SYSTEM '*' COUNT
 (2)  4432  00  00                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4434  00                                 .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4435  07                                 .BYTE   7               ;REQUEST CODE
 (1)  4436  3A  9A  4F     13.0          LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4439  3D              4.0          DCR     A                       ;DOWNCOUNT
 (1)  443A  32  9A  4F     13.0          STA     ITERA                   ;SAVE COUNT
 (1)  443D  F2  24  43     10.0          JP      TST01                   ;DO TEST UNTIL TILL = 0
1545
```

```
                                .SBTTL  TEST 2 - PHASE LOCKED LOOP TEST PART 2
                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PHASE LOCKED LOOP TEST PART 2
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST ISSUES A LOOP WRITE TO READ WITH
                                ;*WITH THE PHASE LOCKED _OOP MODULES (M8950)
                                ;*ENABLED.  IF ANY ERRORS OCCUR, AN ATTEMPT
                                ;*IS MADE TO ISOLATE THE FAILING CHANNEL
                                ;*I.E. M8950 AT FAULT
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  CLEAR ALL TEMP RAM
                                ;*  CLEAR XFRCTL SO WE USE PI L'S
                                ;*  ISSUE A LOOP WRITE TO READ IN GCR
                                ;*  IF ANY ERRORS OCCURED
                                ;*  :  THEN
                                ;*  :    READ ALL TRACK TIE REGISTERS AND
                                ;*  :    PRINT TRACKS AT FAULT
                                ;*  :  ELSE
                                ;*  :    CONTINUE
                                ;*  ENDIF
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM6 MICRO TEST 02
                                ;*RPM6 MICRO ERROR 01
                                ;*RPM6-PHASE LOCKED LOOPS TEST PART 2
                                ;*M8950 CHANNEL (N)
                                ;*LOOP WRITE / READ ERROR WITH PLL ENABLED
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT - LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT   NN
                                ;*ACTUAL DATA IS CHANNEL (N) TIE BUS REGISTFR
                                S
                                ; ************************************************************************
```

```
1583  4440                                          TEST2:  TESTX   @2                      ;INITIALIZE TEST NUMBER 2
 (1)  4440  3E  02                7.0                        MVI     A,@2                    ;DEFINE THE TEST NUMBER
 (1)  4442  CD  03  28           18.0                        CALL    TSET                    ;SETUP THE TEST
1584                                            ;%RPM6-PHASE LOCKED LOOP TEST PART 2
1585                                            ;&M8950 CHANNEL (N)
1586  4445  AF                    4.0  TST02:  XRA     A                       ;CLEAR A
1587  4446  32  09  49            13.0         STA     WFFLG                   ;ALLOW WRITE FAIL BIT CHECKING
1588  4449  32  07  49            13.0         STA     XFRCTL                  ;ALLOW PLL TO BE ENABLED
1589  444C  CD  34  47            18.0         CALL    GCRSUB                  ;SET UP FOR GCR LWR
1590  444F  CD  EA  47            18.0         CALL    LWR
1591  4452  D2  11  45            10.0         JNC     ENDT2                   ;IF NO ERROR GO TO NEXT TEST
1592  4455  3E  03                 7.0         MVI     A,R.STNM+R.STPC         ;STOP READ PATH SO WE CAN
1593                                                                            ;READ REGISTERS 20-27 , 40-47
1594  4457  D3  09                10.0         OUT     RPCTL
1595  4459  AF                    4.0         XRA     A                       ;
1596  445A                                      ROUT    EDATA                   ;WRITE EXPECTED DATA
 (1)  445A  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  445C  7F                     4.0                 MOV     A,A             ;RETRY LINK
1597  445D  DB  20                10.0         IN      CHOTIE                  ;READ CH 0 TIE BUS REG
1598  445F  A7                     4.0         ANA     A                       ;ANY ERROR BITS
1599  4460  CA  6B  44            10.0         JZ      CHOCO2                  ;NO , CONTINUE
1600  4463                                      ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4463  D3  94                10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4465  7F                     4.0                 MOV     A,A             ;RETRY LINK
1601  4466                                      ERRB    TEST2,CHOCO2,3
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4466  CD  12  28           18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000C                                           MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4469  0C                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  446A  03                                             .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  446B  CD  15  28           18.0                CHOCO2::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  446E  DA  40  44           10.0                        JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1602                                            ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1603                                            ;<CHANNEL 0 FAILED
1604                                            ;<ACTUAL DATA IS CHANNEL 0 TIE BUS REGISTER
1605  4471  DB  21                10.0         IN      CH1TIE                  ;READ CH 1 TIE BUS REG
1606  4473  A7                     4.0         ANA     A                       ;ANY ERROR BITS
1607  4474  CA  7F  44            10.0         JZ      CH1CO2                  ;NO , CONTINUE
1608  4477                                      ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4477  D3  94                10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4479  7F                     4.0                 MOV     A,A             ;RETRY LINK
1609  447A                                      ERRB    TEST2,CH1CO2,3
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  447A  CD  12  28           18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000D                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447D  0D                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447E  03                                             .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  447F  CD  15  28           18.0                CH1CO2::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4482  DA  40  44           10.0                        JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1610                                            ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1611                                            ;<CHANNEL 1 FAILED
1612                                            ;<ACTUAL DATA IS CHANNEL 1 TIE BUS REGISTER
```

```
1614  4485  DB  22        10.0          IN      CH2TIE                  ;READ CH 2 TIE BUS REG
1615  4487  A7            4.0           ANA     A                       ;ANY ERROR BITS
1616  4488  CA  93  44    10.0          JZ      CH2CO2                  ;NO , CONTINUE
1617  4488                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4488  D3  94         10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  448D  7F             4.0                   MOV     A,A             ;RETRY LINK
1618  448E                               ERRB    TEST2,CH2CO2,3
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  448E  CD  12  28    18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4491  0E                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4492  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4493  CD  15  28    18.0          CH2CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4496  DA  40  44    10.0                  JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1619                                   ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1620                                   ;<CHANNEL 2 FAILED
1621                                   ;<ACTUAL DATA IS CHANNEL 2 TIE BUS REGISTER
1622  4499  DB  23        10.0          IN      CH3TIE                  ;READ CH 3 TIE BUS REG
1623  4498  A7            4.0           ANA     A                       ;ANY ERROR BITS
1624  449C  CA  A7  44    10.0          JZ      CH3CO2                  ;NO , CONTINUE
1625  449F                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  449F  D3  94         10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44A1  7F             4.0                   MOV     A,A             ;RETRY LINK
1626  44A2                               ERRB    TEST2,CH3CO2,3
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  44A2  CD  12  28    18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A5  0F                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A6  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44A7  CD  15  28    18.0          CH3CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44AA  DA  40  44    10.0                  JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1627                                   ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1628                                   ;<CHANNEL 3 FAILED
1629                                   ;<ACTUAL DATA IS CHANNEL 3 TIE BUS REGISTER
```

```
1631  44AD  DB  24          10.0        IN      CH4TIE                  ;READ CH 4 TIE BUS REG
1632  44AF  A7              4.0         ANA     A                       ;ANY ERROR BITS
1633  44B0  CA  B8  44      10.0        JZ      CH4CO2                  ;NO , CONTINUE
1634  44B3                              ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44B3  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44B5  7F              4.0                 MOV     A,A             ;RETRY LINK
1635  44B6                              ERRB    TEST2,CH4CO2,3
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA 'EDATA" VALID
 (1)  44B6  CD  12  28      18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0010                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B9  10                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44BA  03                              .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44BB  CD  15  28      18.0        CH4CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44BE  DA  40  44      10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1636                                ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1637                                ;<CHANNEL 4 FAILED
1638                                ;<ACTUAL DATA IS CHANNEL 4 TIE BUS REGISTER
1639  44C1  DB  25          10.0        IN      CH5TIE                  ;READ CH 5 TIE BUS REG
1640  44C3  A7              4.0         ANA     A                       ;ANY ERROR BITS
1641  44C4  CA  CF  44      10.0        JZ      CH5CO2                  ;NO , CONTINUE
1642  44C7                              ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44C7  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44C9  7F              4.0                 MOV     A,A             ;RETRY LINK
1643  44CA                              ERRB    TEST2,CH5CO2,3
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA 'EDATA" VALID
 (1)  44CA  CD  12  28      18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0011                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CD  11                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44CE  03                              .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44CF  CD  15  28      18.0        CH5CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D2  DA  40  44      10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1644                                ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1645                                ;<CHANNEL 5 FAILED
1646                                ;<ACTUAL DATA IS CHANNEL 5 TIE BUS REGISTER
```

```
1648  44D5  DB  26          10.0        IN      CH6TIE                  ;READ CH 6 TIE BUS REG
1649  44D7  A7               4.0        ANA     A                       ;ANY ERROR BITS
1650  44D8  CA  E3  44       10.0        JZ      CH6CO2                  ;NO , CONTINUE
1651  44DB                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44DB  D3  94            10.0              OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44DD  7F                 4.0              MOV     A,A             ;RETRY LINK
1652  44DE                               ERRB    TEST2,CH6CO2,3
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44DE  CD  12  28       18.0              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0012                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E1  12                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E2  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44E3  CD  15  28       18.0        CH6CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44E6  DA  40  44       10.0              JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1653                                ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1654                                ;<CHANNEL 6 FAILED
1655                                ;<ACTUAL DATA IS CHANNEL 6 TIE BUS REGISTER
1656  44E9  DB  27          10.0        IN      CH7TIE                  ;READ CH 7 TIE BUS REG
1657  44EB  A7               4.0        ANA     A                       ;ANY ERROR BITS
1658  44EC  CA  F7  44       10.0        JZ      CH7CO2                  ;NO , CONTINUE
1659  44EF                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44EF  D3  94            10.0              OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44F1  7F                 4.0              MOV     A,A             ;RETRY LINK
1660  44F2                               ERRB    TEST2,CH7CO2,3
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44F2  CD  12  28       18.0              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0013                              MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F5  13                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F6  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44F7  CD  15  28       18.0        CH7CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FA  DA  40  44       10.0              JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1661                                ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1662                                ;<CHANNEL 7 FAILED
1663                                ;<ACTUAL DATA IS CHANNEL 7 TIE BUS REGISTER
```

```
1665  44FD  DB  28          10.0        IN      CHPTIE                  ;READ PARITY TIE BUS REG
1666  44FF  A7                4.0        ANA     A                       ;ANY ERROR BITS
1667  4500  CA  0B  45       10.0        JZ      CHPCO2                  ;NO , CONTINUE
1668  4503                              ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4503  D3  94            10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4505  7F                 4.0                MOV     A,A             ;RETRY LINK
1669  4506                              ERRB    TEST2,CHPCO2,3
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4506  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0014                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4509  14                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450A  03                               .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  450B  CD  15  28       18.0        CHPCO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  450E  DA  40  44       10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1670                                    ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1671                                    ;<PARITY CHANNEL FAILED
1672                                    ;<ACTUAL DATA IS PARITY CHANNEL TIE BUS REGISTER
1673  4511                              ENDT2:  ENDTST  TST02
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4511                                      REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4511  CD  06  28       18.0                CALL    REQST
 (2)  4514  00                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  4515  00  00                           .WORD                   ;SYSTEM '*' COUNT
 (2)  4517  00  00                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4519  00                               .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  451A  07                               .BYTE   7               ;REQUEST CODE
 (1)  451B  3A  9A  4F       13.0        LDA     ITERA               ;GET ITERATION COUNT
 (1)  451E  3D                4.0        DCR     A                   ;DOWNCOUNT
 (1)  451F  32  9A  4F       13.0        STA     ITERA               ;SAVE COUNT
 (1)  4522  F2  45  44       10.0        JP      TST02               ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 3 - WRITE FAIL LINES CLEAR TEST
                               ST
                               ;*************************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*WRITE FAIL LINES CLEAR TEST
                               SD
                               ;*************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS TEST PERFORMS A GCR LOOP WRITE TO READ
                               ;*AND THEN CHECKS TO SEE THAT ALL WRITE FAIL LINES
                               ;*ARE CLEAR
                               SP
                               ;*************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*  GET USER SELECTED TU PORT
                               ;*  SET THE RECORD SIZE TO 55.
                               ;*  BGNDO
                               ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
                               ;*  :  SET THE READ PATH CLOCK TO NORMAL
                               ;*  :  CLEAR ALL AMTIE LINES
                               ;*  :  CLEAR TEMP REGISTER SAVE LOCATIONS
                               ;*  :  CALL GCRSUB
                               ;*  :  IF ANY WRITE FAIL LINES ARE SET
                               ;*  :  :  THEN
                               ;*  :  :  :  PRINT ERROR
                               ;*  :  :  ELSE
                               ;*  :  :  :  CONTINUE
                               ;*  :  ENDIF
                               ;*  ENDDO
                               ;*ENDTST
                               SE
                               ;*************************************************************************
                               ;*ERRORS
                               ;*------
                               ;*RPM6 MICRO TEST 03
                               ;*RPM6 MICRO ERROR 01
                               ;*RPM6-WRITE FAIL LINES CLEAR TEST
                               ;*M8950'S , M8953
                               ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                               ;*FATAL ERROR - TEST ABORTED
                               ;*
                               ;*RPM6 MICRO TEST 03
                               ;*RPM6 MICRO ERROR 02
                               ;*RPM6-WRITE FAIL LINES CLEAR TEST
                               ;*M8950'S,M8953
                               ;*WRITE FAIL BITS NOT CLEAR
                               ;*ACTUAL = AAAA
                               ;*EXPECTED = EEEE
                               ;*ACTUAL DATA IS WRITE FAIL REGISTER
                               ;*
```

```
1717                                    ;*RPM6 MICRO TEST 03
1718                                    ;*RPM6 MICRO ERROR 03
1719                                    ;*RPM6-WRITE FAIL LINES CLEAR TEST
1720                                    ;*M8950 , M8953
1721                                    ;*WRITE FAIL PARITY BIT NOT CLEAR
1722                                    ;*ACTUAL = AAAA
1723                                    ;*EXPECTED = EEEE
1724                                    ;*
1725                                    ;*RPM6 MICRO TEST 03
1726                                    ;*RPM6 MICRO ERROR 04
1727                                    ;*RPM6-WRITE FAIL LINES CLEAR TEST
1728                                    ;*M8950'S , M8953
1729                                    ;*LOOP WRITE/READ ERROR
1730                                    ;*ACTUAL = AAAA
1731                                    ;*EXPECTED = EEEE
1732                                    ;*BYTE-SCLK COUNT = LLL
1733                                    ;*DATA FORMAT = MM
1734                                    ;*SKIP COUNT = NN
1735                                    ;*ACTUAL DATA IS THE FAILURE CODE
1736                                    ;*EXPECTED DATA IS THE INTERUPT CODE
1737  4525                              S
 (1)                                    ; ***************************************************************************************
1738
1739  4525                              TEST3:  TESTX   @3                      ;INITIALIZE THE TEST
 (1)  4525  3E  03              7.0                 MVI     A,@3                    ;DEFINE THE TEST NUMBER
 (1)  4527  CD  03  28         18.0                 CALL    TSET                    ;SETUP THE TEST
1740                                    ;%RPM6-WRITE FAIL LINES CLEAR TEST
1741                                    ;&M8950'S , M8953
1742  452A  AF                 4.0     TST03:  XRA     A                       ;CLEAR A
1743  452B  32  08  49        13.0             STA     PLLFLG                  ;CLEAR PLL TEST FLAG
1744  452E  32  07  49        13.0             STA     XFRCTL                  ;STORE PLL CONTROL BYTE
1745  4531  32  0C  49        13.0             STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1746  4534  32  0D  49        13.0             STA     SRPATH                  ;
1747  4537  32  FE  48        13.0             STA     TSTAMT                  ;CLEAR AMTIE CONTROL BYTE
1748  453A  32  FF  48        13.0             STA     TSTAMP                  ;CLEAR AMTIE PARITY CONTROL BYTE
1749  453D  3E  FF              7.0             MVI     A,0FF                   ;SET WRITE FAIL TESTING FLAG
1750  453F  32  09  49        13.0             STA     WFFLG                   ;TO PREVENT LWR FROM TESTING WRITE FAIL
1751  4542  CD  34  47        18.0             CALL    GCRSUB                  ;CALL SUBROUTINE LWRSUB
```

```
1753  4545  21  0C  49      10.0          LXI     H,SRPFAL                ;GET RPFAIL CONTENTS
1754  4548  7E                 7.0          MOV     A,M                     ;GET WRITE FAIL BITS
1755  4549  E6  FF             7.0          ANI     0FF                     ;ANY BITS SET ?
1756  454B  CA  5A  45       10.0          JZ      T3ACON                  ;NO , CONTINUE
1757  454E                                 ROUT    ADATA                   ;SAVE ACTUAL DATA
 (1)  454E  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4550  7F                  4.0                  MOV     A,A             ;RETRY LINK
1758  4551  AF                 4.0          XRA     A                       ;CLEAR A
1759  4552                                 ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  4552  D3  95             10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4554  7F                  4.0                  MOV     A,A             ;RETRY LINK
1760  4555                                 ERRB    TEST3,T3ACON
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4555  CD  12  28       18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0015                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4558  15                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4559  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  455A  CD  15  28       18.0                  T3ACON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  455D  DA  25  45       10.0                          JC      TEST3           ;LOOP ADDRESS IF LOOP SPECIFIED
1761                                 ;>WRITE FAIL BITS NOT CLEAR
1762                                 ;<ACTUAL DATA IS WRITE FAIL REGISTER
1763  4560  21  0D  49      10.0    TST03B: LXI     H,SRPATH                ;GET WRITE FAIL PARITY BIT
1764  4563  7E                 7.0          MOV     A,M                     ;
1765  4564  E6  01             7.0          ANI     01                      ;IS IT SET
1766  4566  CA  75  45       10.0          JZ      T3AC01                  ;NO , GOOD JUMP
1767  4569                                 ROUT    ADATA                   ;STORE ACTUAL DATA
 (1)  4569  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  456B  7F                  4.0                  MOV     A,A             ;RETRY LINK
1768  456C  AF                 4.0          XRA     A                       ;MAKE EXPECTED BYTE
1769  456D                                 ROUT    EDATA                   ;STORE IN CAS
 (1)  456D  D3  95             10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  456F  7F                  4.0                  MOV     A,A             ;RETRY LINK
1770  4570                                 ERRB    TEST3,T3AC01
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  4570  CD  12  28       18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0016                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4573  16                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4574  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4575  CD  15  28       18.0                  T3AC01::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4578  DA  25  45       10.0                          JC      TEST3           ;LOOP ADDRESS IF LOOP SPECIFIED
1771                                 ;>WRITE FAIL PARITY BIT NOT CLEAR
1772  457B                           TST03   ENDTST
 (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  457B                                      REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  457B  CD  06  28       18.0                          CALL    REQST
 (2)  457E  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  457F  00  00                                        .WORD                   ;SYSTEM '*' COUNT
 (2)  4581  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4583  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4584  07                                            .BYTE   7               ;REQUEST CODE
 (1)  4585  3A  9A  4F       13.0          LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4588  3D                  4.0          DCR     A                       ;DOWNCOUNT
 (1)  4589  32  9A  4F       13.0          STA     ITERA                   ;SAVE COUNT
 (1)  458C  F2  2A  45       10.0          JP      TST03                   ;DO TEST UNTIL TILL - C
```

```
1774                                .SBTTL  TEST 4 - WRITE FAIL LINES SET TEST
1775  458F                         ST
 (1)                                ;*****************************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1776                                ;*WRITE FAIL LINES SET TEST
1777  458F                         SD
 (1)                                ;*****************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1778                                ;*THIS TEST PERFORMS A GCR LOOP WRITE TO READ
1779                                ;*WHILE SETTING AN AMTIE BIT FOR EACH CHANNEL
1780                                ;*THEN CHECKING THAT THE APPROPRIATE WRITE FAIL
1781                                ;*BIT IS SET
1782  458F                         SP
 (1)                                ;*****************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1783                                ;*BGNTST
1784                                ;*  GET USER SELECTED TU PORT
1785                                ;*  BGNDO
1786                                ;*  :  BGNDO UNTIL ALL CHANNELS HAVE BEEN TESTED
1787                                ;*  :  :  SET AMTIE BIT FOR CHANNEL (N)
1788                                ;*  :  :  CALL SUBROUTINE GCRSUB
1789                                ;*  :  :  IF THE WRITE FAIL BIT FOR CHANNEL (N) IS SET
1790                                ;*  :  :  :  THEN
1791                                ;*  :  :  :     CONTINUE
1792                                ;*  :  :  :  ELSE
1793                                ;*  :  :  :     PRINT ERROR
1794                                ;*  :  :  :     CONTINUE
1795                                ;*  :  :  ENDIF
1796                                ;*  :  ENDDO
1797                                ;*  ENDDO
1798                                ;*ENDTST
1799  458F                         SE
 (1)                                ;*****************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1800                                ;*RPM6 MICRO TEST 04
1801                                ;*RPM6 MICRO ERROR 01
1802                                ;*RPM6-WRITE FAIL LINES SET TEST
1803                                ;*M8950 CHANNEL (N), M8953
1804                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1805                                ;*FATAL ERROR - TEST ABORTED
1806                                ;*
1807                                ;*RPM6 MICRO TEST 04
1808                                ;*RPM6 MICRO ERROR 02
1809                                ;*RPM6-WRITE FAIL LINES SET TEST
1810                                ;*M8950 CHANNEL (N),M8953
1811                                ;*WRITE FAIL LINE FOR CHANNEL (N) NOT SET
1812                                ;*ACTUAL = AAAA
1813                                ;*EXPECTED = EEEE
1814                                ;*
1815                                ;*RPM6 MICRO TEST 04
```

```
1816                                    ;*RPM6 MICRO ERROR 03
1817                                    ;*RPM6-WRITE FAIL LINES SET TEST
1818                                    ;*M8950 CHANNEL (N),M8953
1819                                    ;*LOOP WRITE/READ ERROR
1820                                    ;*ACTUAL = AAAA
1821                                    ;*EXPECTED = EEEE
1822                                    ;*BYTE-SCLK COUNT = LLL
1823                                    ;*DATA FORMAT = MM
1824                                    ;*SKIP COUNT = NN
1825                                    ;*ACTUAL DATA IS THE FAILURE CODE
1826                                    ;*EXPECTED DATA IS THE INTERUPT CODE
1827  458F                              S
 (1)                                    ; ***************************************************************************
1828  458F                              TEST4:  TESTX   @4                      ;INITIALIZE TEST
 (1)  458F  3E  04            7.0                MVI     A,@4                ;DEFINE THE TEST NUMBER
 (1)  4591  CD  03  28       18.0                CALL    TSET                ;SETUP THE TEST
1829                                    ;%RPM6-WRITE FAIL LINES SET TEST
1830                                    ;&M8950 CHANNEL (N),M8953
1831  4594  AF                 4.0     TST04A: XRA     A                       ;CLEAR A
1832  4595  32  0C  49          13.0             STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1833  4598  32  0D  49          13.0             STA     SRPATH                  ;
1834  459B  32  FF  48          13.0             STA     TSTAMP                  ;CLEAR PARITY AMTIE LINE
1835  459E  3E  01               7.0             MVI     A,@001                  ;GET BIT FOR AMTIE CHANNEL 0
1836  45A0  32  FE  48          13.0             STA     TSTAMT                  ;SET AMTIE CHANNEL 0
1837  45A3  CD  34  47          18.0             CALL    GCRSUB                  ;DO LOOP WRITE / READ
1838  45A6  21  0C  49          10.0             LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1839  45A9  7E                   7.0             MOV     A,M                     ;
1840  45AA  E6  01               7.0             ANI     @001                    ;IS THE WRITE FAIL BIT SET
1841                                                                             ;FOR CHANNEL 0
1842  45AC  C2  BC  45          10.0             JNZ     T4ACON                  ;YES , GOOD CONTINUE
1843  45AF                             ROUT    ADATA                           ;STORE ACTUAL DATA IN CAS
 (1)  45AF  D3  94            10.0                OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  45B1  7F                 4.0                MOV     A,A                 ;RETRY LINK
1844  45B2  3E  01               7.0     MVI     A,@001                          ;
1845  45B4                             ROUT    EDATA                           ;STORE EXPECTED DATA
 (1)  45B4  D3  95            10.0                OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  45B6  7F                 4.0                MOV     A,A                 ;RETRY LINK
1846  45B7                             ERRB    TST04A,T4ACON
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  45B7  CD  12  28       18.0                CALL    ERLPB               ;PROCESS ERROR - DO 2.3
 (1)        0017                             MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45BA  17                               .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  45BB  00                               .BYTE                           ;PRINT ROUTINE NUMBER
 (1)  45BC  CD  15  28       18.0        T4ACON::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45BF  DA  94  45       10.0                JC      TST04A              ;LOOP ADDRESS IF LOOP SPECIFIED
1847                                    ;>WRITE FAIL LINE FOR CHANNEL 0 NOT SET
```

```
1849  45C2  AF              4.0  TST04B: XRA     A                       ;CLEAR A
1850  45C3  32  0C  49       13.0         STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1851  45C6  32  0D  49       13.0         STA     SRPATH                  ;
1852  45C9  3E  02            7.0         MVI     A,@002                  ;
1853  45CB  32  FE  48       13.0         STA     TSTAMT                  ;SET AMTIE CHANNEL 1
1854  45CE  CD  34  47       18.0         CALL    GCRSUB                  ;DO LOOP WRITE / READ
1855  45D1  21  0C  49       10.0         LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1856  45D4  7E                7.0         MOV     A,M                     ;
1857  45D5  E6  02            7.0         ANI     @002                    ;IS THE WRITE FAIL BIT SET
1858                                                                      ;FOR CHANNEL 1
1859  45D7  C2  E7  45       10.0         JNZ     T4BCON                  ;YES , GOOD CONTINUE
1860  45DA                                 ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  45DA  D3  94           10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45DC  7F                4.0                 MOV     A,A             ;RETRY LINK
1861  45DD  3E  02            7.0         MVI     A,@002                  ;
1862  45DF                                 ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  45DF  D3  95           10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45E1  7F                4.0                 MOV     A,A             ;RETRY LINK
1863  45E2                                 ERRB    TST04B,T4BCON
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  45E2  CD  12  28       18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0018                               MSGN    =       MSGN+1   ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45E5  18                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45F6  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45E7  CD  15  28       18.0         T4BCON::        CALL    CKLOP   ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45EA  DA  C2  45       10.0                 JC      TST04B          ;LOOP ADDRESS IF LOOP SPECIFIED
1864                             ;>WRITE FAIL LINE FOR CHANNEL 1 NOT SET
1865  45ED  AF              4.0  TST04C: XRA     A                       ;CLEAR A
1866  45EE  32  0C  49       13.0         STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1867  45F1  32  0D  49       13.0         STA     SRPATH                  ;
1868  45F4  3E  04            7.0         MVI     A,@004                  ;
1869  45F6  32  FE  48       13.0         STA     TSTAMT                  ;SET AMTIE CHANNEL 2
1870  45F9  CD  34  47       18.0         CALL    GCRSUB                  ;DO LOOP WRITE / READ
1871  45FC  21  0C  49       10.0         LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1872  45FF  7E                7.0         MOV     A,M                     ;
1873  4600  E6  04            7.0         ANI     @004                    ;IS THE WRITE FAIL BIT SET
1874                                                                      ;FOR CHANNEL 2
1875  4602  C2  12  46       10.0         JNZ     T4CCON                  ;YES , GOOD CONTINUE
1876  4605                                 ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  4605  D3  94           10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4607  7F                4.0                 MOV     A,A             ;RETRY LINK
1877  4608  3E  04            7.0         MVI     A,@004                  ;
1878  460A                                 ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  460A  D3  95           10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  460C  7F                4.0                 MOV     A,A             ;RETRY LINK
1879  460D                                 ERRB    TST04C,T4CCON
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  460D  CD  12  28       18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0019                               MSGN    -       MSGN+1   ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4610  19                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4611  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4612  CD  15  28       18.0         T4CCON::        CALL    CKLOP   ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4615  DA  ED  45       10.0                 JC      TST04C          ;LOOP ADDRESS IF LOOP SPECIFIED
1880                             ;>WRITE FAIL LINE FOR CHANNEL 2 NOT SET
```

```
1882  4618  AF                    4.0  TST04D: XRA     A                       ;CLEAR A
1883  4619  32  0C  49           13.0          STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1884  461C  32  0D  49           13.0          STA     SRPATH                  ;
1885  461F  3E  08                7.0          MVI     A,@010                  ;
1886  4621  32  FE  48           13.0          STA     TSTAMT                  ;SET AMTIE CHANNEL 3
1887  4624  CD  34  47           18.0          CALL    GCRSUB                  ;DO LOOP WRITE / READ
1888  4627  21  0C  49           10.0          LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1889  462A  7E                    7.0          MOV     A,M                     ;
1890  462B  E6  08                7.0          ANI     @010                    ;IS THE WRITE FAIL BIT SET
1891                                                                           ;FOR CHANNEL 3
1892  462D  C2  3D  46           10.0          JNZ     T4DCON                  ;YES , GOOD CONTINUE
1893  4630                                     ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  4630  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4632  7F                    4.0                  MOV     A,A             ;RETRY LINK
1894  4633  3E  08                7.0          MVI     A,@010                  ;
1895  4635                                     ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  4635  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4637  7F                    4.0                  MOV     A,A             ;RETRY LINK
1896  4638                                     ERRB    TST04D,T4DCON
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4638  CD  12  28           18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)        001A                                      MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  463B  1A                                        .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  463C  00                                        .BYTE                     ;PRINT ROUTINE NUMBER
 (1)  463D  CD  15  28           18.0          T4DCON::        CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4640  DA  18  46           10.0                  JC      TST04D            ;LOOP ADDRESS IF LOOP SPECIFIED
1897                                     ;>WRITE FAIL LINE FOR CHANNEL 3 NOT SET
1898  4643  AF                    4.0  TST04E: XRA     A                       ;CLEAR A
1899  4644  32  0C  49           13.0          STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1900  4647  32  0D  49           13.0          STA     SRPATH                  ;
1901  464A  3E  10                7.0          MVI     A,@020                  ;
1902  464C  32  FE  48           13.0          STA     TSTAMT                  ;SET AMTIE CHANNEL 4
1903  464F  CD  34  47           18.0          CALL    GCRSUB                  ;DO LOOP WRITE / READ
1904  4652  21  0C  49           10.0          LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1905  4655  7E                    7.0          MOV     A,M                     ;
1906  4656  E6  10                7.0          ANI     @020                    ;IS THE WRITE FAIL BIT SET
1907                                                                           ;FOR CHANNEL 4
1908  4658  C2  68  46           10.0          JNZ     T4ECON                  ;YES , GOOD CONTINUE
1909  465B                                     ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  465B  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  465D  7F                    4.0                  MOV     A,A             ;RETRY LINK
1910  465E  3E  10                7.0          MVI     A,@020                  ;
1911  4660                                     ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  4660  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4662  7F                    4.0                  MOV     A,A             ;RETRY LINK
1912  4663                                     ERRB    TST04E,T4ECON
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4663  CD  12  28           18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)        001B                                      MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4666  1B                                        .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  4667  00                                        .BYTE                     ;PRINT ROUTINE NUMBER
 (1)  4668  CD  15  28           18.0          T4ECON::        CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  466B  DA  43  46           10.0                  JC      TST04E            ;LOOP ADDRESS IF LOOP SPECIFIED
1913                                     ;>WRITE FAIL LINE FOR CHANNEL 4 NOT SET
```

```
1915  466E  AF                  4.0  TST04F: XRA     A                       ;CLEAR A
1916  466F  32  0C  49          13.0          STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1917  4672  32  0D  49          13.0          STA     SRPATH                  ;
1918  4675  3E  20               7.0          MVI     A,@040                  ;
1919  4677  32  FE  48          13.0          STA     TSTAMT                  ;SET AMTIE CHANNEL 5
1920  467A  CD  34  47          18.0          CALL    GCRSUB                  ;DO LOOP WRITE / READ
1921  467D  21  0C  49          10.0          LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1922  4680  7E                   7.0          MOV     A,M                     ;
1923  4681  E6  20               7.0          ANI     @040                    ;IS THE WRITE FAIL BIT SET
1924                                                                          ;FOR CHANNEL 5
1925  4683  C2  93  46          10.0          JNZ     T4FCON                  ;YES , GOOD CONTINUE
1926  4686                                    ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  4686  D3  94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4688  7F                   4.0                  MOV     A,A             ;RETRY LINK
1927  4689  3E  20               7.0          MVI     A,@040                  ;
1928  468B                                    ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  468B  D3  95               10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  468D  7F                   4.0                  MOV     A,A             ;RETRY LINK
1929  468E                                    ERRB    TST04F,T4FCON
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  468E  CD  12  28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001C                                    MSGN    =       MSGN    ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4691  1C                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4692  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4693  CD  15  28          18.0          T4FCON::        CALL    CKLOP   ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4696  DA  6E  46          10.0                  JC      TST04F          ;LOOP ADDRESS IF LOOP SPECIFIED
1930                                    ;>WRITE FAIL LINE FOR CHANNEL 5 NOT SET
1931  4699  AF                  4.0  TST04G: XRA     A                       ;CLEAR A
1932  469A  32  0C  49          13.0          STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1933  469D  32  0D  49          13.0          STA     SRPATH                  ;
1934  46A0  3E  40               7.0          MVI     A,@100                  ;
1935  46A2  32  FE  48          13.0          STA     TSTAMT                  ;SET AMTIE CHANNEL 6
1936  46A5  CD  34  47          18.0          CALL    GCRSUB                  ;DO LOOP WRITE / READ
1937  46A8  21  0C  49          10.0          LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1938  46AB  7E                   7.0          MOV     A,M                     ;
1939  46AC  E6  40               7.0          ANI     @100                    ;IS THE WRITE FAIL BIT SET
1940                                                                          ;FOR CHANNEL 6
1941  46AE  C2  BE  46          10.0          JNZ     T4GCON                  ;YES , GOOD CONTINUE
1942  46B1                                    ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  46B1  D3  94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46B3  7F                   4.0                  MOV     A,A             ;RETRY LINK
1943  46B4  3E  40               7.0          MVI     A,@100                  ;
1944  46B6                                    ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  46B6  D3  95               10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46B8  7F                   4.0                  MOV     A,A             ;RETRY LINK
1945  46B9                                    ERRB    TST04G,T4GCON
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  46B9  CD  12  28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001D                                    MSGN    =       MSGN+1  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46BC  1D                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46BD  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46BE  CD  15  28          18.0          T4GCON::        CALL    CKLOP   ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46C1  DA  99  46          10.0                  JC      TST04G          ;LOOP ADDRESS IF LOOP SPECIFIED
1946                                    ;>WRITE FAIL LINE FOR CHANNEL 6 NOT SET
```

```
1948  46C4  AF                    4.0  TST04H: XRA     A                       ;CLEAR A
1949  46C5  32  0C  49            13.0         STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1950  46C8  32  0D  49            13.0         STA     SRPATH                  ;
1951  46CB  3E  80                 7.0         MVI     A,@200                  ;
1952  46CD  32  FE  48            13.0         STA     TSTAMT                  ;SET AMTIE CHANNEL 7
1953  46D0  CD  34  47            18.0         CALL    GCRSUB                  ;DO LOOP WRITE / READ
1954  46D3  21  0C  49            10.0         LXI     H SRPFAL                ;GET WRITE FAIL LINES
1955  46D6  7E                     7.0         MOV     A,M                     ;
1956  46D7  E6  80                 7.0         ANI     @200                    ;IS THE WRITE FAIL BIT SET
1957                                                                           ;FOR CHANNEL 7
1958  46D9  C2  E9  46            10.0         JNZ     T4HCON                  ;YES , GOOD CONTINUE
1959  46DC                                     ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  46DC  D3  94                10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46DE  7F                     4.0                 MOV     A,A             ;RETRY LINK
1960  46DF  3E  80                 7.0         MVI     A,@200                  ;
1961  46E1                                     ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  46E1  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46E3  7F                     4.0                 MOV     A,A             ;RETRY LINK
1962  46E4                                     ERRB    TST04H,T4HCON
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46E4  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001E                                       MSGN    =       MSGN+1  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E7  1E                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46E8  00                                         .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46E9  CD  15  28            18.0         T4HCON::        CALL    CKLOP   ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46EC  DA  C4  46            10.0                 JC      TST04H          ;LOOP ADDRESS IF LOOP SPECIFIED
1963                                     ;>WRITE FAIL LINE FOR CHANNEL 7 NOT SET
```

```
 1965  46EF  AF                    4.0  TST04I: XRA     A                       ;CLEAR A
 1966  46F0  32  0C  49            13.0         STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
 1967  46F3  32  0D  49            13.0         STA     SRPATH                  ;
 1968  46F6  32  FE  48            13.0         STA     TSTAMT                  ;CLEAR AMTIES
 1969  46F9  3E  01                 7.0         MVI     A,@001                  ;
 1970  46FB  32  FF  48            13.0         STA     TSTAMP                  ;SET AMTIE PARITY LINE
 1971  46FE  CD  34  47            18.0         CALL    GCRSUB                  ;DO LOOP WRITE / READ
 1972  4701  21  0D  49            10.0         LXI     H,SRPATH                ;GET PARITY WRITE FAIL LINE
 1973  4704  7E                     7.0         MOV     A,M                     ;
 1974  4705  E6  01                 7.0         ANI     @001                    ;IS THE WRITE FAIL BIT SET
 1975                                                                          ;FOR THE PARITY LINE
 1976  4707  C2  17  47            10.0         JNZ     T4ICON                  ;YES , GOOD CONTINUE
 1977  470A                                     ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
  (1)  470A  D3  94                10.0                 OUT     ADATA          ;WRITE AC INTO ADATA
  (1)  470C  7F                     4.0                 MOV     A,A            ;RETRY LINK
 1978  470D  3E  01                 7.0         MVI     A,@001                  ;
 1979  470F                                     ROUT    EDATA                   ;STORE EXPECTED DATA
  (1)  470F  D3  95                10.0                 OUT     EDATA          ;WRITE AC INTO EDATA
  (1)  4711  7F                     4.0                 MOV     A,A            ;RETRY LINK
 1980  4712                                     ERRB    TST04I,T4ICON
  (1)                                        ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
  (1)  4712  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        001F                                       MSGN    =       MSGN+1  ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4715  1F                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4716  0C                                         .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4717  CD  15  28            18.0         T4ICON::        CALL    CKLOP   ;CHECK LOOP FUNCTION - DO 2.2
  (1)  471A  DA  EF  46            10.0                 JC      TST04I          ;LOOP ADDRESS IF LOOP SPECIFIED
 1981                                       ;>WRITE FAIL LINE FOR PARITY NOT SET
 1982  471D                                     ENDTST  TST04A
  (1)                                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  471D                                     REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
  (2)  471D  CD  06  28            18.0                         CALL    REQST
  (2)  4720  00                                                 .BYTE           ;DATA PATTERN NUMBER
  (2)  4721  00  00                                             .WORD           ;SYSTEM '*' COUNT
  (2)  4723  00  00                                             .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4725  00                                                 .BYTE           ;DATA COMPARE FLAG IF =1
  (2)  4726  07                                                 .BYTE   7       ;REQUEST CODE
  (1)  4727  3A  9A  4F            13.0                 LDA     ITERA           ;GET ITERATION COUNT
  (1)  472A  3D                     4.0                 DCR     A               ;DOWNCOUNT
  (1)  472B  32  9A  4F            13.0                 STA     ITERA           ;SAVE COUNT
  (1)  472E  F2  94  45            10.0                 JP      TST04A          ;DO TEST UNTIL TILL = 0
 1983  4731  C3  18  28            10.0  EXIT:          JMP     TSTEND
 1984
```

```
1986                                    .SBTTL  SUBROUTINE GCRSUB
1987
1988                                    ;*SUBROUTINE: GCRSUB
1989                                    ;*
1990                                    ;*DESCRIPTION:
1991                                    ;*
1992                                    ;*      THIS SUBROUTINE INITIALIZES ALL VARIABLES IN ORDER
1993                                    ;*      TO RUN THE LOOP WRITE / READ CODE
1994                                    ;*
1995                                    ;*      SET TU PORT UNDER TEST
1996                                    ;*      CLEAR ALL TU PORTS
1997                                    ;*      SET RECORD SIZE IN CAS
1998                                    ;*      SET FORMAT IN CAS
1999                                    ;*      LOAD DATA PATTERN IN DDR
2000                                    ;*      CALL LWR
2001                                    ;*
2002                                    ;*VARIABLES:
2003                                    ;*      PLLFLG -- IF SET TO NON ZERO WILL NOT ALLOW LWR TO BE CALLED
2004                                    ;*
2005                                    ;*SUBROUTINES CALLED:
2006                                    ;*      CLEAR
2007                                    ;*      LWR
2008
2009  4734  06  00          7.0  GCRSUB: MVI     B,@0
2010  4736  3A  00  49      13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
2011  4739  E6  01           7.0         ANI     @01             ;PORT 0?
2012  473B  C2  51  47      10.0         JNZ     FOUND           ;YES-GO USE PORT #0
2013  473E  04               4.0         INR     B               ;NO-UPDATE POINTER TO PORT #1
2014  473F  3A  00  49      13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
2015  4742  E6  02           7.0         ANI     @02             ;PORT 1?
2016  4744  C2  51  47      10.0         JNZ     FOUND           ;YES-GO USE PORT #1
2017  4747  04               4.0         INR     B               ;NO-UPDATE POINTER TO PORT #2
2018  4748  3A  00  49      13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
2019  474B  E6  04           7.0         ANI     @04             ;PORT 2?
2020  474D  C2  51  47      10.0         JNZ     FOUND           ;YES-GO USE PORT #2
2021  4750  04               4.0         INR     B               ;NO-ASSUME PORT #3
2022  4751  CD  BF  47      18.0 FOUND:  CALL    CLEAR
2023  4754  DB  E0          10.0         IN      INTSTA
2024  4756  E6  80           7.0         ANI     BIT7
2025  4758  B0               4.0         ORA     B
2026  4759  D3  E0          10.0         OUT     MBSEL           ;SELECT THE DESIRED TU-PORT AND MASSBUS
2027
2028  475B  AF               4.0 T1L01:  XRA     A               ;CLEAR THE DATA BUS CONTROL
2029  475C  32  28  4F      13.0         STA     DBUS            ;MASK TO KEEP THE MASSBUS CLEAN
2030                                                            ;
2031  475F  2A  01  49      16.0         LHLD    SIZE            ;GET THE RECORD SIZE
2032  4762  22  23  4F      16.0         SHLD    BYTEL
2033  4765  7D               4.0         MOV     A,L             ;GET THE BYTE COUNT LOW
2034  4766                               ROUT    RO5L            ;SAVE IN BYTE COUNT LOW
  (1) 4766  D3  8A          10.0                 OUT     RO5L    ;WRITE AC INTO RO5L
  (1) 4768  7F               4.0                 MOV     A,A     ;RETRY LINK
2035  4769  7C               4.0         MOV     A,H             ;GET THE BYTE COUNT HIGH
2036  476A                               ROUT    RO5H            ;SAVE IN BYTE COUNT HIGH
  (1) 476A  D3  8B          10.0                 OUT     RO5H    ;WRITE AC INTO RO5H
```

```
  (1)  476C  7F                    4.0          MOV     A,A             ;RETRY LINK
2037                                             ;
2038   476D  3A  03  49            13.0  LDA     FORMT                   ;GET THE FORMAT
2039   4770  32  25  4F            13.0  STA     FORMAT                  ;NO SKIP COUNT
2040   4773                             ROUT    R02H                    ;SAVE IN THE CAS
  (1)  4773  D3  85                10.0          OUT     R02H            ;WRITE AC INTO R02H
  (1)  4775  7F                    4.0          MOV     A,A             ;RETRY LINK
2041                                             ;
```

```
2043  4776  3E  88        7.0          MVI   A,DDRC0     ;CLEAR DDR IN, SO DDR
2044  4778  D3  DB       10.0          OUT   DDRCTL      ;WRITES TO TAPE
2045                                                     ;
2046  477A  2A  04  49    16.0          LHLD  KWDAT       ;GET THE WRITE DATA
2047  477D  29            10.0          DAD   H           ;SHIFT IT LEFT 1 BIT POSITION
2048  477E  17             4.0          RAL               ;SAVE THE CARRY
2049  477F  29            10.0          DAD   H           ;SHIFT IT LEFT ANOTHER BIT POSITION
2050  4780  17             4.0          RAL               ;SAVE THAT CARRY TOO.
2051  4781  E6  03         7.0          ANI   $3          ;SAVE ONLY THE CARRY BITS
2052  4783  47             4.0          MOV   B,A         ;SAVE IN REGISTER B
2053  4784  7C             4.0          MOV   A,H         ;
2054  4785  D3  D9        10.0          OUT   DDRB        ;LOAD <13:6> IN DDR B
2055  4787  3A  06  49    13.0          LDA   KWDAT+2     ;GET BITS <17:16>
2056  478A  85             4.0          ADD   L                 ;MERGE WITH <5:0>
2057  478B  D3  D8        10.0          OUT   DDRA        ;LOAD INTO DDR A
2058  478D  AC             4.0          XRA   H           ;COMPUTE PARITY ON ALL BITS
2059  478E  A8             4.0          XRA   B           ;
2060  478F  78             4.0          MOV   A,B         ;GET BITS <15:14> AGAIN
2061  4790  E2  95  47     10.0          JPO   1$          ;JUMP IF ODD PARITY
2062  4793  F6  04         7.0          ORI   BIT2        ;SET THE PARITY BIT TO MAKE IT ODD
2063  4795  D3  DA        10.0    1$:   OUT   DDRC        ;LOAD <P:14> IN DDRC
2064                                                     ;
2065  4797  21  08  49    10.0          LXI   H,PLLFLG    ;GET PHASE LOCKED LOOP FLAG
2066  479A  7E             7.0          MOV   A,M         ;
2067  479B  A7             4.0          ANA   A           ;ARE WE DOING THE PLL TEST
2068  479C  C2  BE  47     10.0          JNZ   QUIT        ;YES , EXIT AND LET PLL TEST CONTROL
2069                                                     ;THE REST
2070  479F  CD  EA  47     18.0          CALL  LWR         ;CALL LOOP WRITE/READ - TU PORT - GCR
2071                                                     ;
2072  47A2  D2  B8  47     10.0    3$:   JNC   T1CN1       ;CONTINUE IF NO ERROR
2073  47A5  78             4.0          MOV   A,B         ;GET THE FAILURE CODE
2074  47A6  E6  FC         7.0          ANI   $FC         ;MASK OUT THE MEANINGFUL BITS
2075  47A8  0F             4.0          RRC               ;POSITION FOR OUTPUT
2076  47A9  0F             4.0          RRC               ;
2077  47AA                              ROUT  ADATA       ;STORE IN THE CAS
 (1)  47AA  D3  94        10.0                OUT   ADATA  ;WRITE AC INTO ADATA
 (1)  47AC  7F             4.0                MOV   A,A    ;RETRY LINK
2078  47AD  79             4.0          MOV   A,C         ;GET THE INTERRUPT CODE
2079  47AE  E6  3F         7.0          ANI   $3F         ;MASK OUT THE MEANINGFUL BITS
2080  47B0                              ROUT  EDATA       ;STORE IN THE CAS
 (1)  47B0  D3  95        10.0                OUT   EDATA  ;WRITE AC INTO EDATA
 (1)  47B2  7F             4.0                MOV   A,A    ;RETRY LINK
2081  47B3                              ERRB  T1LO1,T1CN1,3
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  47B3  CD  12  28    18.0                CALL  ERLPB       ;PROCESS ERROR - DO 2.3
 (1)       0020                               MSGN  =     MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47B6  20                                .BYTE MSGN      ;MESSAGE NUMBER ID
 (1)  47B7  03                                .BYTE 3         ;PRINT ROUTINE NUMBER
 (1)  47B8  CD  15  28    18.0          T1CN1:: CALL  CKLOP   ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47BB  DA  5B  47    10.0                JC    T1LO1     ;LOOP ADDRESS IF LOOP SPECIFIED
2082                              ;>LOOP WRITE/READ ERROR
2083                              ;<ACTUAL DATA IS THE FAILURE CODE
2084                              ;<EXPECTED DATA IS THE INTERRUPT CODE
2085  47BE  C9        10.0    QUIT: RET
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
47BF                            SSUB
 (1)                            ;*************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*CLEAR ALL TU PORTS
47BF                            SD
 (1)                            ;*************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
47BF                            SP
 (1)                            ;*************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*      SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*      CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*      CLEAR PORT SELECT FOR TRANSPORT
                                ;*      CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*      CLEAR PORT DIAGNOSTIC CONTROL
                                ;*      CLEAR PORT AMTIE WORD
                                ;*ENDSUB
47BF                            S
 (1)                            ;*************************************************************************

47BF  F5            12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
47C0  C5            12.0          PUSH    B
47C1  06  00         7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
47C3  DB  E0        10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
47C5  E6  80         7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
47C7  B0             4.0          ORA     B               ;ADD IN THE SELECTED PORT #
47C8  D3  E0        10.0          OUT     MBSEL           ;RESET TO THIS PORT
47CA  3E  80         7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
47CC  D3  40        10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
47CE  AF             4.0          XRA     A               ;CLEAR TU COMMAND A
47CF  D3  40        10.0          OUT     TCMD            ;
47D1  3E  81         7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
47D3  D3  40        10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
47D5  3E  00         7.0          MVI     A,SELCLR        ;LOAD TU ''CLEAR SELECT'' COMMAND
47D7  D3  40        10.0          OUT     TCMD            ;ISSUE TU ''CLEAR SELECT'' FOR TRANSPORT #0
47D9  AF             4.0          XRA     A
47DA  D3  44        10.0          OUT     TAMT            ;CLEAR AMTIE WORD
47DC  D3  48        10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
47DE  D3  4C        10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
47E0  04             4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
47E1  78             4.0          MOV     A,B
47E2  FE  04         7.0          CPI     4               ;DONE?
47E4  C2  C3  47    10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
47E7  C1            10.0          POP     B               ;RESET B & C
47E8  F1            10.0          POP     PSW             ;ALL DONE
47E9  C9            10.0          RET                     ;EXIT
```

```
                                        .SBTTL  LWR - GCR LOOP WRITE TO READ AT PORT BOARD

                                ;*SUBROUTINE: LWR - GCR LOOP WRITE TO READ AT PORT BOARD
                                ;*
                                ;*DESCRIPTION:
                                ;*
                                ;*      THIS SUBROUTINE SETS UP THE TM78 HARDWARE REGISTERS
                                ;*      TO LOOP THE WRITE PATH TO THE READ PATH THROUGH
                                ;*      THE M8955 (TU78 PORT MODULE)
                                ;*      THEN A GCR DATA TRANSFER IS ISSUED
                                ;*      ALL STATUS REGISTERS ARE CHECKED FOR VALID
                                ;*      TERMINATION DATA
                                ;*      IF ANY ERRORS OCCUR THIS SUBROUTINE
                                ;*      WILL RETURN WITH THE "C" BIT SET
                                ;*      AND AN ERROR CODE IN THE REGISTER PAIR B ANN C
                                ;*
                                ;*VARIABLES:
                                ;*      WFFLG -- IF SET TO NON ZERO DISABLES WRITE FAIL CHECKING
                                ;*
                                ;*SUBROUTINES CALLED:
                                ;*      WRTSET -- SET UP WRITE PATH (LOCATED IN OPERATIONAL MICROCODE)
                                ;*
                                ;*


47EA  3E  21          7.0  LWR:   MVI   A,P.AMTP+P.LWR          ;SET LWR AT PORT AND AMTIE PAR BIT
47EC  D3  48         10.0         OUT   PDIAG

                                ;SET THE AMTIES TO SIMULATE IN A GAP BETWEEN RECORDS

47EE  3E  FF          7.0         MVI   A,0FF                  ;SET ALL AMTIE BITS
47F0  D3  44         10.0         OUT   TAMT
47F2  3E  21          7.0         MVI   A,P.LWR+P.AMTP         ;INCLUDING THE PARITY LINE
47F4  D3  48         10.0         OUT   PDIAG                  ;WRITE IT OUT
                                ;INIT THE WMC

47F6  AF              4.0         XRA   A                      ;CLEAR TRACK ENABLE REGISTER
47F7  D3  D2         10.0         OUT   TRKENA                 ; TO COMPLETELY RESET
47F9  D3  D2         10.0         OUT   TRKENA                 ; THE WRITE PATH
47FB  CD  0C  08     18.0         CALL  WRTSET                 ;SET UP ALL WMC REGISTERS
47FE  3E  0D          7.0         MVI   A,RCLRT                ;ISSUE CLEAR ALL COMMAND TO RMC
4800  D3  0B         10.0         OUT   RCMD                   ;TO PUT THE PLO'S AT REST
                                                               ;EARLY, TO SIMULATE TAPE START TIME

4802  3A  07  49     13.0         LDA   XFRCTL                 ;GET TRANSFER CONTROL WORD
4805  D3  09         10.0  LWR1:  OUT   RPCTL                 ;LOAD RMC CONTROL REGISTER

                                ;START WRITING

4807  3E  14          7.0         MVI   A,P.WPEN+P.RPEN        ;SET WRITE AND READ ENABLE
4809  D3  4C         10.0  LWR2:  OUT   PENAB
480B  3E  D8          7.0         MVI   A,W.ENAB+X.ENAB+W.WRITE+W.GCR
480D  D3  D3         10.0         OUT   WMCCTL                 ;ENABLE THE WMC AND XMC
```

```
2186
2187                                    ;START READ PATH READING
2188
2189  480F  3E  07          7.0         MVI     A,READG             ;SETUP RMC READ COMMAND
2190  4811  D3  0B         10.0         OUT     RCMD
2191
2192                                    ;LOAD AMTIE BITS
2193
2194  4813  3A  FE  48      13.0         LDA     TSTAMT              ;GET AMTIE WORD
2195  4816  D3  44          10.0         OUT     TAMT
2196  4818  3A  FF  48      13.0         LDA     TSTAMP              ;GET AMTIE PARITY BIT
2197  481B  F6  20           7.0         ORI     P.LWR               ;KEEP LWR BIT
2198  481D  D3  48          10.0         OUT     PDIAG               ;WRITE OUT PORT REGISTER
2199                                    ;WATCH FOR XMC DONE
2200                                    ;TIME OUT AFTER 332.8 MS.
2201
2202  481F  01  8C  57      10.0         LXI     B,22412.            ;CLEAR B,C. SET OPERATION TIMER
2203
2204  4822  DB  D0          10.0  LWR8A:  IN      WMCSTA              ;GET WMC,XMC DONE WORD
2205  4824  A7             4.0         ANA     A                   ;MANIPULATE CONDITION CODES
2206                                                                ;SEE IF XMC IS DONE (AC = POSITIVE)
2207  4825  F2  3D  48      10.0         JP      LWR9                ;YES! WRITE DATA XFR IS FINISHED!
2208  4828  0B             6.0         DCX     B                   ;DECREMENT LOOP COUNTER
2209  4829  78             4.0         MOV     A,B                 ;LOOP UNTIL DONE OR TIMED OUT
2210  482A  B1             4.0         ORA     C
2211  482B  C2  22  48      10.0         JNZ     LWR8A               ;LOOP UNTIL DONE OR TIMED OUT
2212
2213                                    ;IF WE GET HERE THE READ-AFTER-WRITE HAS TIMED OUT
2214                                    ;QUIT!
2215
2216  482E  01  18  48      10.0         LXI     B,I30+F22           ;SET READ-AFTER-WRITE TIME OUT INTERRUPT CODE
2217  4831  DB  E0          10.0         IN      INTSTA
2218  4833  1F             4.0         RAR
2219  4834  D2  B1  48      10.0         JNC     LWREX
2220  4837  01  18  1C      10.0         LXI     B,I30+F07
2221  483A  C3  B1  48      10.0         JMP     LWREX               ;ERROR! GO STOP MOTION.
2222
2223                                    ;XMC IS FINISHED. WRITE DATA XFR IS DONE.
2224
2225                                    ;SET ALL AMTIE BITS TO SATISFY RMC
2226
2227  483D  3E  05           7.0  LWR9:   MVI     A,5                 ;DELAY TO ALLOW READ PATH TO
2228  483F  3D             4.0  LWR9L:  DCR     A                   ; DETECT LONG POSTAMBLE ERROR
2229  4840  C2  3F  48      10.0         JNZ     LWR9L               ; AND ALLOW PLO'S TO DRIFT
2230  4843  3E  FF           7.0         MVI     A,0FF               ;SET ALL AMTIE BITS
2231  4845  D3  44          10.0         OUT     TAMT                ;BITS <7:0>
2232  4847  3E  21           7.0         MVI     A,P.LWR+P.AMTP  ;SET PARITY AMITE BIT
2233  4849  D3  48          10.0         OUT     PDIAG               ;WRITE
2234
2235  484B  3E  00           7.0         MVI     A,0                 ;DELAY ABOUT 1 MS. TO ALLOW
2236  484D  3D             4.0  LWR10:  DCR     A                   ;  READ PATH TO TIME OUT
2237  484E  C2  4D  48      10.0         JNZ     LWR10               ;  0.1 IN. GAP AND FINISH
2238                                    ;CHECK ERROR BITS
2239
```

```
2240  4851  3E  9B        7.0          MVI   A,DDRCO+DDRAIN+DDRBIN+DDRCIN    ;SET DDR IN
2241  4853  D3  DB       10.0          OUT   DDRCTL                          ;SO M.RDPE CAN BE READ IN WMCERR REG
2242  4855  DB  DA       10.0          IN    WMCERR                          ;CHECK WMC ERROR BITS
2243  4857  32  0A  49   13.0          STA   SWMCER                          ;COPY REGISTER TO MEMORY
2244  485A  E6  38        7.0          ANI   W.ERR+W.ROME+M.RDPE
2245  485C  C2  BC  48   10.0          JNZ   LWRER2                          ;ERROR IF ANY SET
2246
2247                                ;CHECK IF FINAL RMC STATUS IS ACCEPTABLE
2248
2249  485F  DB  02       10.0          IN    RSTAT                           ;READ RMC STATUS WORD
2250  4861  32  0B  49   13.0          STA   SRSTAT                          ;STORE IN MEMORY
2251  4864  FE  B1        7.0          CPI   PL                              ;CHECK IF OKAY
2252  4866  DA  B6  48   10.0          JC    LWRER1                          ;IF NOT, ERROR
2253
2254
2255  4869  21  09  49   10.0          LXI   H,WFFLG          ;WRITE FAIL TEST ?
2256  486C  7E           7.0          MOV   A,M                             ;GET THE FLAG
2257  486D  A7           4.0          ANA   A                               ;SET CONDITION CODE
2258  486E  CA  83  48   10.0          JZ    LWRCHK                          ;NO , DO THIS ERROR CHECKING
2259
2260                                ;WRITE FAIL TEST FLAG IS SET USE THIS CODE
2261  4871  DB  00       10.0          IN    RPFAIL                          ;LOOK AT WRITE FAIL BITS
2262  4873  32  0C  49   13.0          STA   SRPFAL                          ;STORE IN RAM
2263  4876  DB  01       10.0          IN    RPATH                           ;LOOK AT WRITE FAIL PARITY BIT
2264  4878  32  0D  49   13.0          STA   SRPATH                          ;STORE IN RAM
2265  487B  DB  1A       10.0          IN    ECCSTA                          ;CHECK ECC STATUS BITS
2266  487D  32  0E  49   13.0          STA   SECCST                          ;STORE IN RAM
2267  4880  C3  9E  48   10.0          JMP   LWRXX                           ;EXIT NORMALLY
2268
2269                                ;NORMAL LOOP WRITE TO READ CODE
2270
2271  4883  DB  00       10.0  LWRCHK: IN    RPFAIL                          ;LOOK AT WRITE FAIL BITS
2272  4885  32  0C  49   13.0          STA   SRPFAL                          ;STORE IN RAM
2273  4888  A7           4.0          ANA   A                               ;
2274  4889  C2  CA  48   10.0          JNZ   LWRER3                          ;ERROR IF NOT ZERO
2275  488C  DB  01       10.0          IN    RPATH                           ;LOOK AT WRITE FAIL PARITY BIT
2276  488E  32  0D  49   13.0          STA   SRPATH                          ;STORE IN RAM
2277  4891  1F           4.0          RAR                                   ;SHIFT TO "C" BIT
2278  4892  DA  CA  48   10.0          JC    LWRER3                          ;ERROR IF NOT ZERO
2279
2280                                ;CHECK ERROR BITS FOR GCR
2281
2282  4895  DB  1A       10.0  LWRNXA: IN    ECCSTA                          ;CHECK ECC STATUS BITS
2283  4897  32  0E  49   13.0          STA   SECCST                          ;STORE IN RAM
2284  489A  A7           4.0          ANA   A                               ;LOOK FOR ANY ERROR BITS
2285  489B  C2  EA  48   10.0          JNZ   LWRER7                          ;ERROR IF ANY SET
2286
2287                                ;NO ERROR EXIT
2288
2289  489E  DB  DA       10.0  LWRXX:  IN    WMCERR                          ;READ MASSBUS PARITY ERROR BIT
2290  48A0  32  0A  49   13.0          STA   SWMCER                          ;STA IN RAM
2291  48A3  E6  40        7.0          ANI   M.PE
2292  48A5  C2  F8  48   10.0          JNZ   LWRER8                          ;JUMP IF SET
2293  48A8  3E  84        7.0          MVI   A,P.INTEN+P.RPEN                ;CLEAR DATA PATH ENABLE
```

```
2294  48AA  D3 4C       10.0          OUT   PENAB
2295  48AC  01 01 00    10.0          LXI   B,101          ;SET JP DONE INTERRUPT CODE
2296  48AF  AF          4.0           XRA   A              ;CLEAR CARRY
2297  48B0  C9          10.0          RET
2298
2299                               ;ERROR EXIT
2300
2301  48B1  AF          4.0   LWREX: XRA   A
2302  48B2  D3 4C       10.0          OUT   PENAB
2303  48B4  37          4.0           STC                  ;SET CARRY TO INDICATE ERROR
2304  48B5  C9          10.0          RET
2305                               ;ERROR EXITS
2306
2307  48B6  01 15 20    10.0  LWRER1: LXI   B,I25+F10      ;RSTAT CONTAINS ERROR CODE
2308  48B9  C3 B1 48    10.0          JMP   LWREX
2309
2310  48BC  01 18 4C    10.0  LWRER2: LXI   B,I30+F23      ;WMC ROM PE OR RD PE SET IN WMCERR
2311  48BF  E6 20       7.0           ANI   W.ERR          ;REGISTER
2312  48C1  CA B1 48    10.0          JZ    LWREX
2313  48C4  01 18 CC    10.0          LXI   B,I30+F03
2314  48C7  C3 B1 48    10.0          JMP   LWREX
2315
2316  48CA  01 15 18    10.0  LWRER3: LXI   B,I25+F06      ;AT LEAST ONE WRITE FAIL BIT IS SET IN RPFAIL
2317  48CD  C3 B1 48    10.0          JMP   LWREX          ;OR RPATH REGISTERS
2318
2319  48D0  01 15 10    10.0  LWRER4: LXI   B,I25+F04      ;AMTIE, PNTR MISMATCH, UNCORRECTABLE, 2 TRK ERR
2320  48D3  E6 40       7.0           ANI   E.RPE          ;OR SINGLE TRACK ERROR SET IN ECCSTA REGISTER
2321  48D5  CA B1 48    10.0          JZ    LWREX
2322  48D8  01 18 18    10.0          LXI   B,I30+F06
2323  48DB  C3 B1 48    10.0          JMP   LWREX
2324
2325  48DE  01 18 44    10.0  LWRER5: LXI   B,I30+F21      ;XL PE SET IN INTSTA REGISTER
2326  48E1  C3 B1 48    10.0          JMP   LWREX
2327
2328  48E4  01 15 24    10.0  LWRER6: LXI   B,I25+F11      ;CRC CHARACTERS FROM WMC AND RMC DID NOT MATCH
2329  48E7  C3 B1 48    10.0          JMP   LWREX
2330
2331  48EA  01 15 14    10.0  LWRER7: LXI   B,I25+F05      ;AT LEAST ONE BIT SET IN ECCSTA REGISTER
2332  48ED  E6 40       7.0           ANI   E.RPE
2333  48EF  CA B1 48    10.0          JZ    LWREX
2334  48F2  01 18 18    10.0          LXI   B,I30+F06
2335  48F5  C3 B1 48    10.0          JMP   LWREX
2336
2337  48F8  01 15 28    10.0  LWRER8: LXI   B,I25+F12      ;MASSBUS DATA BUS PARITY ERROR
2338  48FB  C3 B1 48    10.0          JMP   LWREX
2339
2340                               .SBTTL  PROGRAM VARIABLES
2341
2342  48FE  00                     TSTAMT: .BYTE  0         ;AMTIE BYTE TO LOAD INTO TAMT
2343  48FF  00                     TSTAMP: .BYTE  0         ;AMTIE PARITY BYTE
2344  4900  00                     UNITMP: .BYTE  0         ;UNIT MAP
2345  4901  00 00                  SIZE:   .WORD  0         ;RECORD SIZE
2346  4903  00                     FORMT:  .BYTE  0         ;DATA FORMAT
2347  4904  00                     KWDAT:  .BYTE  0         ;18 BITS OF INPUT DATA
```

```
2348  4905  00                        .BYTE   0
2349  4906  00                        .BYTE   0
2350  4907  00              XFRCTL:   .BYTE   0             ;RPCTL SOFTWARE REGISTER
2351  4908  00              PLLFLG:   .BYTE   0             ;PHASE LOCKED LOOP TEST FLAG
2352  4909  00              WFFLG:    .BYTE   0             ;WRITE FAIL TEST FLAG
2353
2354                        ;TEMPORARY REGISTER SAVE LOCATIONS
2355
2356  490A  00              SWMCER:   .BYTE   0             ;WMCERR SAVE LOCATION
2357  490B  00              SRSTAT:   .BYTE   0             ;RSTAT SAVE LOCATION
2358  490C  00              SRPFAL:   .BYTE   0             ;RPFAIL SAVE LOCATION
2359  490D  00              SRPATH:   .BYTE   0             ;RPATH SAVE LOCATION
2360  490E  00              SECCST:   .BYTE   0             ;ECCSTA SAVE LOCATION
2361        0000            .END
```

RPM6.M80 SYMBOL TABLE

```
A       %0007           ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098            ASAVE    4F9B          ATTCD    4F97         AXNUM    4F91
B      =%0000           BADST  = 0090          BIT0   = 0001         BIT1   = 0002
BIT15 = 8000            BIT2   = 0004          BIT3   = 0008         BIT4   = 0010
BIT5   = 0020           BIT6   = 0040          BIT7   = 0080         BIT8   = 0100
BIT9   = 0200           BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00            BSAVE    4F9C          BYTCNT= 00D4          BYTEH    4F24
BYTEL    4F23           C      =%0001          CASCT    4F21         CASCTL= 00A0
CASSTA= 00A0            CATTH  = 0089          CATTL  = 0088         CBUSST= 00A1
CBYTH  = 0088           CBYTL  = 008A          CDG1H  = 0087         CDG1L  = 0086
CDG2H  = 0093           CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094
CDVTH  = 008D           CDVTL  = 008C          CHPCON   4426  G      CHPCO2   450B  G
CHPTIE= 0028            CH0CON   4386  G       CH0CO2   446B  G      CH0TIE= 0020
CH1CON   439A  G        CH1CO2   447F  G       CH1TIE= 0021          CH2CON   43AE  G
CH2CO2   4493  G        CH2TIE= 0022           CH3CON   43C2  G      CH3CO2   44A7  G
CH3TIE= 0023            CH4CON   43D6  G       CH4CO2   44BB  G      CH4TIE= 0024
CH5CON   43EA  G        CH5CO2   44CF  G       CH5TIE= 0025          CH6CON   43FE  G
CH6CO2   44E3  G        CH6TIE= 0026           CH7CON   4412  G      CH7CO2   44F7  G
CH7TIE= 0027            CKLOP  = 2815          CLEAR    47BF         CLKCTL= 00F0
CLOCK    4F26           CLRLP    47C3          CMC0H  = 0099         CMC0L  = 0098
CMC1H  = 009B           CMC1L  = 009A          CMC2H  = 009D         CMC2L  = 009C
CMC3H  = 009F           CMC3L  = 009E          CMINH  = 0097         CMINL  = 0096
CNTCTL= 00D7            CRCWRD= 0018           CSAVE    4F9D         CSRLH  = 0091
CSRLL  = 0090           CTCH   = 0085          CTCL   = 0084         CTSTH  = 008F
CTSTL  = 008E           CXCTH  = 0081          CXCTL  = 0080         CXINH  = 0083
CXINL  = 0082           C.     = 0001          C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010           C.DTU  = 0003          C.DVA  = 0008         C.FAIL= 00FC
C.FMT  = 0070           C.FNCT= 003E           C.GO   = 0001         C.INTC= 00FE
C.MAIN= 0020            C.NSA  = 0080          C.RCT  = 00FC         C.SER  = 0080
C.SHR  = 0040           C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002
D      =%0002           DATACT= 00D0           DBUS     4F28         DBUSCT= 00C0
DBUSST= 00C0            DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002            DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB            DIAFLG   4F22          DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 00DB           DONE1  = 0045          DONINT= 0010          DSAVE    4F9E
DSE    = 0006           DUMMY    431E  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010            D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008            D.WR4  = 0080          E      =%0003         ECCBAD= 0042
ECCCOR= 0019            ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA  - 0095           ENDT2    4511          EOTCLR= 0003          ERFLG    4F93
ERLP   = 2809           ERLPA  = 280F          EPLPB  = 2812         ERLPE  = 280C
ERNUM    4F90           ERRCNT= 00D6           ESAVE    4F9F         EXIT     4731
E.ACRC= 0010            E.AMT  = 0020          E.CDP  = 0080         E.CRC  = 0080
E.PNTR- 0008            E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004           FIFORD- 006A           FORMAT   4F25         FORMT    4903
FOUND    4751           FWDTST= 0061           F03    = 0C00         F04    = 1000
F05    - 1400           F06    = 1800          F07    = 1C00         F10    = 2000
F11    = 2400           F12    = 2800          F21    = 4400         F22    = 4800
F23    - 4C00           GCRID  = 0089          GCRSET= 0002          GCRSUB   4734
GOODTM= 0092            H      =%0004          HLSAVE   4FA0         IE     = 0008
INTSTA= 00E0            ITERA    4F9A          I.PWR  = 0020         I.RMPE= 0040
I01    = 0001           I25    = 0015          I30    = 0018         I5.5   = 0010
I6.5   = 0020           I7.5   = 0040          KCALL  = 005F         KCLR   - 007B
KDEP   - 003F           KENAB  = 0078          KEXAM  = 003E         KEYBRD= 00C8
```

```
KEY1   = 0078    KEY10  = 006D    KEY11  = 006E    KEY12  = 006F
KEY13  = 005C    KEY14  = 005D    KEY15  = 005E    KEY16  = 005F
KEY17  = 003C    KEY18  = 003D    KFY19  = 003E    KEY2   = 0079
KEY20  = 003F    KEY3   = 007A    KEY4   = 007B    KEY5   = 0074
KEY6   = 0075    KEY7   = 0076    KEY8   = 0077    KEY9   = 006C
KINTA  = 006F    KLDAD  = 003D    KN0    = 003C    KN1    = 005C
KN2    = 005D    KN3    = 005E    KN4    = 006C    KN5    = 006D
KN6    = 006E    KN7    = 0074    KN8    = 0075    KN9    = 0076
KU2    = 007D    KU3    = 007A    KU8    = 0077    KWDAT    4904
L      =%0005    LBLANK= 000F     LCE    = 000B    LCH    = 000C
LCL    = 000D    LCP    - 000E    LC0    = 0000    LC1    = 0001
LC2    = 0002    LC3    = 0003    LC4    = 0004    LC5    = 0005
LC6    = 0006    LC7    = 0007    LC8    = 0008    LC9    = 0009
LDLEDA= 00CA     LDLEDB= 00CB     LDLEDC= 00CC     LDLEDD= 00CD
LDLEDE= 00CE     LDLEDF= 00CF     LKDIAG- 2800     LKKBD  = 004C
LKKEY  = 0049    LKLWMG= 0058     LKLWMP- 0055     LKLWPG= 0052
LKLWPP= 004F     LKMOD7- 0046     LKOPR  = 0046    LPFLG    4F94
LPNUM    4F92    LWR      47EA    LWRCHK   4883    LWRER1   48B6
LWRER2   48BC    LWRER3   48CA    LWRER4   48D0    LWRER5   48DE
LWRER6   48E4    LWRER7   48EA    LWRER8   48F8    LWREX    48B1
LWRNXA   4895    LWRXX    489E    LWR1     4805    LWR10    484D
LWR2     4809    LWR8A    4822    LWR9     483D    LWR9L    483F
M      -%0006    MBSEL  = 00E0    MB.A   - 0008    MB.B   = 0004
MEMTOP- 4FFF     MINUS  = 000A    MM     - 8000    MSE    = 0008
MSGN   - 0020    MTACLR= 0000     MT.ARA- 0020     MT.CPE= 0080
MT.DSE- 0001     MT.FWD- 0040     MT.INH= 0008     MT.LWR= 0004
MT.MOT= 0002     MT.NWT- 0080     MT.PEC- 0040     MT.PSB= 0004
MT.PSO- 0001     MT.PS1= 0002     MT.REV= 0020     MT.WRT= 0010
MT.Z   - 0008    M.ATA  = 0080    M.CAPE- 0020     M.CONT= 0080
M.DEM  = 0020    M.EBL  = 0010    M.EXC  = 0008    M.FAIL= 0008
M.ILR  - 0010    M.INIT= 0010     M.OCC  = 0020    M.ONLI= 0001
M.PE   = 0040    M.PORT= 0080     M.RDEN= 0002     M.RDPE= 0008
M.RUN  - 0004    M.SCLK= 0001     M.TRA  = 0040    M.UNIT= 0007
M.WCLK- 0040     M.WCLN= 0080     M.WREN= 0080     M5,5   = 0001
M6.5   - 0002    M7.5   = 0004    NOTCAP= 0088     OKAY   = 00FF
OPRRAM= 4300     OPSTRT- 0058     OPVER  = 0040    PADCNT= 00D5
PADCRC- 0080     PDIAG  = 0048    PEID   = 008A    PENAB  = 004C
PESET  - 0001    PL     = 00B1    PLLFLG   4908    PRDD   = 004C
PRENF  = 009C    PS     = 00B2    PSTAT  = 0048    PSW    =%0009
P.AMTP= 0001     P.BCTC= 0040     P.CMDP= 0020     P.INTE= 0080
P.LCS  = 0040    P.LWR  = 0020    P.RDP  = 0002    P.RPEN= 0004
P.RPST= 0002     P.RP0E= 0020     P.RP1E= 0010     P.RP2E= 0020
P.RP3E= 0010     P.SING= 0080     P.STAT= 0002     P.STPE= 0080
P.TACH= 0008     P.TUPR= 0010     P.WCSP= 0004     P.WDS  = 0040
P.WFLP= 0001     P.WPEN- 0010     P.WP0E- 0008     P.WP1E= 0004
P.WP2E= 0008     P.WP3E= 0004     P.5VOK= 0002     QUE    = 281B
QUEM   = 281E    QUIT     47BE    RAMT   = 0010    RARA   = 0006
RARAI  = 0004    RCHBD0= 0048     RCHBD1= 0047     RCHOK  = 0046
RCHTST= 000C     RCLRT  = 000D    RCMD   - 000B    RCMLP  = 0003
RCONT  = 0080    RDATA  = 0017    RDCLK  = 0010    RDON   = 0011
READG  = 0007    REND   = 0014    REQST  = 2806    RESCHR= 00D1
REVTST= 0064     REWIND= 0004     RFIFOL= 0008     RGCLK  = 0002
RGCRI  = 0003    RIBG   = 0001    RILL   = 0012    RI'IST = 000C
RMCTST= 0008     RMK2   = 0013    RNOP   - 0000    RPATH  = 0001
```

```
RPBAD = 0044        RPCHI = 0001        RPCLK = 0003        RPCTL = 0009
RPEI  = 0002        RPFAIL= 0000        RPF1  - 009D        RPF2  = 009E
RPOK  = 0043        RPOSTN= 0016        RPSTA = 0015        RRCMT = 000A
RSTAT = 0002        RTIEB = 000A        RTIER = 0030        RTM   = 0005
RUNKI = 0009        RUPTST= 005E        RWDUNL= 0005        R.AMT = 0001
R.BOP = 0008        R.DATA= 0040        R.DON = 0002        R.DRDY= 0010
R.END - 0010        R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008
R.PLOD- 0008        R.PLOO= 0010        R.PLO1= 0020        R.POST= 0020
R.STNM= 0002        R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080
R.TSTD= 0040        R.VOK = 0080        R00H  = 0081        R00L  = 0080
R01H  = 0083        R01L  = 0082        R02H  = 0085        R02L  = 0084
R03H  = 0087        R03L  = 0086        R04H  = 0089        R04L  = 0088
R05H  = 0088        R05L  = 008A        R06H  = 008D        R06L  = 008C
R07H  = 008F        R07L  = 008E        R10H  = 0091        R10L  = 0090
R11H  = 0093        R11L  = 0092        R12H  = 0095        R12L  = 0094
R13H  = 0097        R13L  = 0096        R14H  = 0099        R14L  = 0098
R15H  = 009B        R15L  = 009A        R16H  = 009D        R16L  = 009C
R17H  = 009F        R17L  = 009E        R7.5  = 0010        SECCST  490E
SELCLR= 0000        SETATA= 00A1        SID   = 0080        SIZE    4901
SOD   = 0080        SOE   = 0040        SP    =%0008        SRPATH  490D
SRPFAL  490C        SRSTAT  4908        SSCLK = 0040        SSTEP = 0005
STACK = 4FFF        STATRM= 4F20        STPCT   4F20        STRSP = 5000
SWMCER  490A        TADR00= 0080        TADR01= 0081        TADR02= 0082
TADR03= 0083        TADR04= 0084        TADR05= 0085        TADR06= 0086
TADR07= 0087        TADR10= 0088        TADR11= 0089        TADR12= 008A
TADR13= 008B        TAMT  = 0044        TASEL = 0080        TCMD  = 0040
TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004        TC.REV= 0020
TC.WRT= 0010        TEMP    4F99        TEST1   4300        TEST2   4440
TEST3   4525        TEST4   458F        TMF   = 0099        TMRDY = 0040
TRKENA= 00D2        TSET  = 2803        TSTAMP  48FF        TSTAMT  48FE
TSTEND= 2818        TSTS  = 0040        TST01   4324        TST01A  4370
TST02   4445        TST03   452A        TST03B  4560        TST04A  4594
TST04B  45C2        TST04C  45ED        TST04D  4618        TST04E  4643
TST04F  466E        TST04G  4699        TST04H  46C4        TST04I  46EF
TUSEL0= 00D1        TUSEL1= 00D2        TU78  = 0010        T.ATH0= 0001
T.ATH1= 0002        T.BOT = 0004        T.EOT - 0002        T.FPT = 0001
T.NTHR= 0004        T.ONL = 0020        T.PES = 0008        T.PSBJ= 0020
T.PSOJ- 0008        T.PS1J= 0010        T.RDY = 0080        T.RDYO= 0040
T.RWD - 0010        T.SCLK= 0002        T1CN1   47B8  G     T1CON1  436A  G
T1LO1   475B        T3ACON  455A  G     T3AC01  4575  G     T4ACON  45BC  G
T4BCON  45E7  G     T4CCON  4612  G     T4DCON  463D  G     T4ECON  4668  G
T4FCON  4693  G     T4GCON  468E  G     T4HCON  46E9  G     T4ICON  4717  G
UIBG  - 00A1        UNITMP  4900        VALFC   4F98        VALTB   4F95
VELTST- 00CB        WDR.P = 0010        WFFLG   4909        WMCCTL= 00D3
WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000        WRTDAT= 00D3
WRTSET= 080C        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT - 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X     =%000A
XFRCTL  4907        X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002
X.ROME= 0001        X.WCLK= 0001        Y     =%000B        .     = 490F
```

ERRORS DETECTED: 0

*RPM6.A78/PTP,RPM6=NLIST,PARAM,MACRO,LIST,RPM6
RUN-TIME: 4 6 0 SECONDS
CORE USED: 10K

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(26) 6-FEB-81 18:01 SEQ 1230
PARAM.M80 TABLE OF CONTENTS

```
                                .TITLE  TUP2 - TAPE UNIT PORT TEST PART #2
                                .SBTTL  TEST 1 - LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
                                ;ID     TUP2-TAPE UNIT PORT CONTROLLER PART #2
4300                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR - LOOP WRITE/READ - TU PORT - NORMAL W/R CLOCKS
4300                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH NORMAL WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
4300                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
                                ;*  :  SET THE READ PATH CLOCK TO NORMAL
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE 640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE GCRSUB
                                ;*ENDTST
4300                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 01
                                ;*TUP2 MICRO ERROR 01
                                ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
                                ;*ALL MODULES
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*TUP2 MICRO TEST 01
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT - MM
                                ;*SKIP COUNT - NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
```

```
1371                                    ;*EXPECTED DATA IS THE INTERRUPT CODE
1372  4300                              S
 (1)                                    ; ************************************************************************************
1373
1374  4300  C3  07  43      10.0        JMP     TEST1
1375  4303  00                          .BYTE   0               ;DUMMY BYTE
1376  4304  00                          .BYTE   0               ;DUMMY BYTE
1377  4305  00                          .BYTE   0               ;DUMMY BYTE
1378  4306  00                  SPEED:  .BYTE   0               ;CLOCK CONTROL WORD
1379  4307                      TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4307  3E  01                       7.0                    MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4309  CD  03  28          18.0    CALL    TSET            ;SETUP THE TEST
1380                                    ;%TUP2-LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
1381                                    ;&ALL MODULES
1382  430C                              REQ     @7,0,0,0,0
 (1)  430C  CD  06  28          18.0                    CALL    REQST
 (1)  430F  00                                          .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4310  00  00                                      .WORD   0               ;SYSTEM ''0'' COUNT
 (1)  4312  00  00                                      .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4314  00                                          .BYTE   0               ;DATA COMPARE FLAG IF  1
 (1)  4315  07                                          .BYTE   @7              ;REQUEST CODE
1383  4316                              RIN     R12L
 (1)  4316  DB  94              10.0                    IN      R12L            ;READ R12L INTO AC
 (1)  4318  7F                   4.0                    MOV     A,A             ;RETRY LINK
1384  4319  32  77  45          13.0    STA     UNITMP
1385  431C  A7                   4.0    ANA     A
1386  431D  C2  2B  43          10.0    JNZ     TST01X
1387  4320                              ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4320  CD  09  28          18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4323  01                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4324  00                                          .BYTE
 (1)  4325  CD  15  28          18.0            DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4328  DA  A6  44          10.0                    JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1388                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1389                                    ;>FATAL ERROR - TEST ABORTED
```

```
1391  432B  21  80  00      10.0  TST01X: LXI   H,128                ;SET UP THE BYTE COUNT
1392  432E  22  78  45      16.0          SHLD  SIZE                 ;
1393                                                                 ;
1394  4331  3E  10           7.0  TST01Y: MVI   A,@020               ;SET UP FOR NORMAL CLOCKS
1395  4333  32  06  43      13.0          STA   SPEED
1396  4336  CD  B0  44      18.0          CALL  GCRSUB          ;CALL SUBROUTINE LWRSUB
1397                                                                 ;
1398  4339  2A  78  45      16.0          LHLD  SIZE                 ;GET THE SIZE
1399  433C  01  80  00      10.0          LXI   B,128                ;LOAD THE INCRFMENT VALUE
1400  433F  09              10.0          DAD   B                    ;INCREMENT THE BYTE COUNT
1401  4340  22  78  45      16.0          SHLD  SIZE                 ;SAVE THE SIZE FOR LATER
1402  4343  7C               4.0          MOV   A,H                  ;DO UNTIL THE BYTE COUNT = 768
1403  4344  FE  03           7.0          CPI   $3                   ;
1404  4346  C2  31  43      10.0          JNZ   TST01Y               ;
1405  4349  21  58  04      10.0          LXI   H,1112.              ;SET UP THE BYTE COUNT
1406  434C  22  78  45      16.0          SHLD  SIZE                 ;
1407  434F  CD  B0  44      18.0          CALL  GCRSUB               ;CALL THE LWRSUB SUBROUTINE
1408  4352                                ENDTST TST01X
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4352                              REQ   7                    ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4352  CD  06  28      18.0                          CALL    REQST
 (2)  4355  00                                            .BYTE                ;DATA PATTERN NUMBER
 (2)  4356  00  00                                        .WORD                ;SYSTEM "" COUNT
 (2)  4358  00  00                                        .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  435A  00                                            .BYTE                ;DATA COMPARE FLAG IF  1
 (2)  435B  07                                            .BYTE   7            ;REQUEST CODE
 (1)  435C  3A  9A  4F      13.0          LDA   ITERA                ;GET ITERATION COUNT
 (1)  435F  3D               4.0          DCR   A                    ;DOWNCOUNT
 (1)  4360  32  9A  4F      13.0          STA   ITERA                ;SAVE COUNT
 (1)  4363  F2  2B  43      10.0          JP    TST01X               ;DO TEST UNTIL TILL = 0
1409
```

```
                                .SBTTL  TEST 2 - LOOP WRITE/READ - TU PORT - GCR - +10% W/R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR - LOOP WRITE/READ - TU PORT +10% W/R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNIST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE-640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE GCRSUB
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 02
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT   LLL
                                ;*DATA FORMAT   MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************
```

```
1449  4366                                    TEST02: TESTX   @2               ;INITIALIZE THE TEST
 (1)  4366  3E  02              7.0                     MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4368  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1450                                    ;%TUP2-LOOP WRITE/READ - TU PORT - GCR - +10% W/R CLOCKS
1451                                    ;&ALL MODULES
1452  436B  21  80  00         10.0    TST02:  LXI     H,128           ;SETUP THE BYTE COUNT
1453  436E  22  78  45         16.0            SHLD    SIZE            ;
1454                                                                    ;
1455  4371  3E  0A               7.0    TST02X: MVI     A,@012          ;SET W/R CLOCKS TO +10%
1456  4373  32  06  43         13.0            STA     SPEED           ;
1457  4376  CD  B0  44         18.0            CALL    GCRSUB          ;CALL SUBROUTINE LWRSUB
1458  4379  2A  78  45         16.0            LHLD    SIZE            ;GET THE BYTE COUNT
1459  437C  01  80  00         10.0            LXI     B,128           ;SET UP INCREMENT VALUE
1460  437F  09                 10.0            DAD     B               ;INCREMENT THE BYTE COUNT
1461  4380  22  78  45         16.0            SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1462  4383  7C                  4.0            MOV     A,H             ;DO UNTIL THE BYTE COUNT-1
1463  4384  FE  03               7.0            CPI     $3
1464  4386  C2  71  43         10.0            JNZ     TST02X
1465  4389  21  58  04         10.0            LXI     H,1112.                 ;SET UP THE BYTE COUNT
1466  438C  22  78  45         16.0            SHLD    SIZE                    ;
1467  438F  CD  B0  44         18.0            CALL    GCRSUB                  ;CALL THE LWRSUB SUBROUTINE
1468  4392                                    ENDTST  TST02
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4392                                      REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4392  CD  06  28         18.0                                     CALL    REQST
 (2)  4395  00                                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  4396  00  00                                                      .WORD                   ;SYSTEM '*' COUNT
 (2)  4398  00  00                                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  439A  00                                                          .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  439B  07                                                          .BYTE   7               ;REQUEST CODE
 (1)  439C  3A  9A  4F         13.0                      LDA     ITERA           ;GET ITERATION COUNT
 (1)  439F  3D                  4.0                      DCR     A               ;DOWNCOUNT
 (1)  43A0  32  9A  4F         13.0                      STA     ITERA           ;SAVE COUNT
 (1)  43A3  F2  6B  43         10.0                      JP      TST02           ;DO TEST UNTIL TILL = 0
1469
```

```
                                .SBTTL  TEST 3 - LOOP WRITE/READ - TU PORT - GCR - -20% W -30% R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR - LOOP WRITE/READ - TU PORT - -20%W -30%R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO -20%
                                ;*  :  SET THE READ PATH CLOCK TO -30%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE-640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE GCRSUB
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 03
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - -20% W -30% R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************
```

```
1509  43A6                                    TEST03: TESTX   @3               ;INITIALIZE THE TEST
 (1)  43A6  3E  03                    7.0                MVI     A,@3          ;DEFINE THE TEST NUMBER
 (1)  43A8  CD  03  28               18.0                CALL    TSET          ;SETUP THE TEST
1510                                        ;%TUP2-LOOP WRITE/READ - TU PORT - GCR - -20% W -30% R CLOCKS
1511                                        ;&ALL MODULES
1512  43AB  21  80  00                10.0 TST03:  LXI     H,128            ;SET UP THE BYTE COUNT
1513  43AE  22  78  45                16.0         SHLD    SIZE             ;
1514                                                                         ;
1515  43B1  3E  1B                     7.0 TST03X: MVI     A,@033           ;SET -20% W AND -30% R CLOCKS
1516  43B3  32  06  43                13.0         STA     SPEED            ;
1517  43B6  CD  B0  44                18.0         CALL    GCRSUB           ;CALL SUBROUTINE LWRSUB
1518  43B9  2A  78  45                16.0         LHLD    SIZE             ;GET THE BYTE COUNT
1519  43BC  01  80  00                10.0         LXI     B,128            ;SET UP THE INCREMENT VALUE
1520  43BF  09                        10.0         DAD     B                ;INCREMENT THE BYTE COUNT
1521  43C0  22  78  45                16.0         SHLD    SIZE             ;SAVE THE NEW BYTE COUNT
1522  43C3  7C                         4.0         MOV     A,H              ;DO UNTIL THE BYTE COUNT=
1523  43C4  FE  03                     7.0         CPI     $3
1524  43C6  C2  B1  43                10.0         JNZ     TST03X
1525  43C9  21  58  04                10.0         LXI     H,1112.                  ;SET UP THE BYTE COUNT
1526  43CC  22  78  45                16.0         SHLD    SIZE                     ;
1527  43CF  CD  B0  44                18.0         CALL    GCRSUB                   ;CALL THE LWRSUB SUBROUTINE
1528  43D2                                        ENDTST  TST03
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43D2                                          REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43D2  CD  06  28               18.0                                CALL    REQST
 (2)  43D5  00                                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  43D6  00  00                                                       .WORD                   ;SYSTEM "" COUNT
 (2)  43D8  00  00                                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43DA  00                                                           .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  43DB  07                                                           .BYTE   7               ;REQUEST CODE
 (1)  43DC  3A  9A  4F               13.0                LDA     ITERA        ;GET ITERATION COUNT
 (1)  43DF  3D                        4.0                DCR     A            ;DOWNCOUNT
 (1)  43E0  32  9A  4F               13.0                STA     ITERA        ;SAVE COUNT
 (1)  43E3  F2  AB  43               10.0                JP      TST03        ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 4 - LOOP WRITE/READ - TU PORT - PE - NORMAL W/R CLOCKS

                                S
                                ; ********** **********
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE/READ - TU PORT - NORMAL W/R CLOCKS
                                SD
                                ; ********************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH NORMAL WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ********************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
                                ;*  :  SET THE READ PATH CLOCK TO NORMAL
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE PESUB
                                ;*ENDTST
                                SE
                                ; ********************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 04
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - PE - NORMAL W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT - LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT - NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ********************************************************************************
```

```
1568  43E6                                  TEST04: TESTX   @4               ;INITIALIZE THE TEST
 (1)  43E6  3E  04              7.0                     MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1)  43E8  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1569                                    ;%TUP2-LOOP WRITE/READ - TU PORT - PE - NORMAL W/R CLOCKS
1570                                    ;&ALL MODULES
1571  43EB  21  80  00         10.0  TST04:  LXI     H,128                    ;SET UP THE BYTE COUNT
1572  43EE  22  78  45         16.0           SHLD    SIZE                     ;
1573                                                                           ;
1574  43F1  3E  10               7.0  TST04X: MVI     A,@020                   ;SET UP FOR NORMAL CLOCKS
1575  43F3  32  06  43          13.0           STA     SPEED
1576  43F6  CD  A9  44          18.0           CALL    PESUB            ;CALL SUBROUTINE PESUB
1577                                                                           ;
1578  43F9  2A  78  45          16.0           LHLD    SIZE                     ;GET THE SIZE
1579  43FC  01  80  00          10.0           LXI     B,128                    ;LOAD THE INCREMENT VALUE
1580  43FF  09                  10.0           DAD     B                        ;INCREMENT THE BYTE COUNT
1581  4400  22  78  45          16.0           SHLD    SIZE                     ;SAVE THE SIZE FOR LATER
1582  4403  7C                   4.0           MOV     A,H                      ;DO UNTIL THE BYTE COUNT = 768
1583  4404  FE  03               7.0           CPI     $3                       ;
1584  4406  C2  F1  43          10.0           JNZ     TST04X                   ;
1585  4409  21  58  04          10.0           LXI     H,1112.                  ;SET UP THE BYTE COUNT
1586  440C  22  78  45          16.0           SHLD    SIZE                     ;
1587  440F  CD  A9  44          18.0           CALL    PESUB                    ;CALL SUBROUTINE PESUB
1588  4412                                    ENDTST  TST04
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4412                                      REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4412  CD  06  28         18.0                             CALL    REQST
 (2)  4415  00                                                  .BYTE           ;DATA PATTERN NUMBER
 (2)  4416  00  00                                              .WORD           ;SYSTEM '*' COUNT
 (2)  4418  00  00                                              .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  441A  00                                                  .BYTE           ;DATA COMPARE FLAG IF -1
 (2)  441B  07                                                  .BYTE   7       ;REQUEST CODE
 (1)  441C  3A  9A  4F          13.0           LDA     ITERA            ;GET ITERATION COUNT
 (1)  441F  3D                   4.0           DCR     A                ;DOWNCOUNT
 (1)  4420  32  9A  4F          13.0           STA     ITERA            ;SAVE COUNT
 (1)  4423  F2  EB  43          10.0           JP      TST04            ;DO TEST UNTIL TILL - 0
1589
```

```
                                .SBTTL  TEST 5 - LOOP WRITE/READ - TU PORT - PE - +10% W/R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*----- -----
                                ;*PE - LOOP WRITE/READ - TU PORT +10% W/R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE-640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE PESUB
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 05
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - PE - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED - EEEE
                                ;*BYTE-SCLK COUNT - LLL
                                ;*DATA FORMAT   MM
                                ;*SKIP COUNT - NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************
```

```
1629  4426                                  TEST05: TESTX   @5               ;INITIALIZE THE TEST
 (1)  4426  3E 05                 7.0                MVI     A,@5             ;DEFINE THE TEST NUMBER
 (1)  4428  CD 03 28             18.0                CALL    TSET             ;SETUP THE TEST
1630                                    ;%TUP2-LOOP WRITE/READ - TU PORT - PE - +10% W/R CLOCKS
1631                                    ;&ALL MODULES
1632  442B  21 80 00             10.0  TST05:  LXI     H,128            ;SETUP THE BYTE COUNT
1633  442E  22 78 45             16.0          SHLD    SIZE             ;
1634                                                                     ;
1635  4431  3E 0A                 7.0  TST05X: MVI     A,@012           ;SET W/R CLOCKS TO +10%
1636  4433  32 06 43             13.0          STA     SPEED            ;
1637  4436  CD A9 44             18.0          CALL    PESUB            ;CALL SUBROUTINE PESUB
1638  4439  2A 78 45             16.0          LHLD    SIZE             ;GET THE BYTE COUNT
1639  443C  01 80 00             10.0          LXI     B,128            ;SET UP INCREMENT VALUE
1640  443F  09                   10.0          DAD     B                ;INCREMENT THE BYTE COUNT
1641  4440  22 78 45             16.0          SHLD    SIZE             ;SAVE THE NEW BYTE COUNT
1642  4443  7C                    4.0          MOV     A,H              ;DO UNTIL THE BYTE COUNT-1
1643  4444  FE 03                 7.0          CPI     $3
1644  4446  C2 31 44             10.0          JNZ     TST05X
1645  4449  21 58 04             10.0          LXI     H,1112.                  ;SET UP THE BYTE COUNT
1646  444C  22 78 45             16.0          SHLD    SIZE                     ;
1647  444F  CD A9 44             18.0          CALL    PESUB                    ;CALL SUBROUTINE PESUB
1648  4452                                    ENDTST  TST05
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4452                                  REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4452  CD 06 28             18.0                        CALL    REQST
 (2)  4455  00                               .BYTE                    ;DATA PATTERN NUMBER
 (2)  4456  00 00                            .WORD                    ;SYSTEM '*' COUNT
 (2)  4458  00 00                            .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  445A  00                               .BYTE                    ;DATA COMPARE FLAG IF -1
 (2)  445B  07                               .BYTE   7                ;REQUEST CODE
 (1)  445C  3A 9A 4F             13.0          LDA     ITERA            ;GET ITERATION COUNT
 (1)  445F  3D                    4.0          DCR     A                ;DOWNCOUNT
 (1)  4460  32 9A 4F             13.0          STA     ITERA            ;SAVE COUNT
 (1)  4463  F2 2B 44             10.0          JP      TST05            ;DO TEST UNTIL TILL = 0
1649
```

```
                                .SBTTL  TEST 6 - LOOP WRITE/READ - TU PORT - PE - -20% W -30% R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE/READ - TU PORT - -20%W -30%R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO -20%
                                ;*  :  SET THE READ PATH CLOCK TO -30%
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE-640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE PESUB
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 06
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - PE - -20% W -30% R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED - EEEE
                                ;*BYTE-SCLK COUNT - LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************
```

```
1689  4466                                  TEST06: TESTX   @6               ;INITIALIZE THE TEST
 (1)  4466  3E  06              7.0                      MVI     A,@6         ;DEFINE THE TEST NUMBER
 (1)  4468  CD  03  28         18.0                      CALL    TSET         ;SETUP THE TEST
1690                                   ;%TUP2-LOOP WRITE/READ - TU PORT - PE - -20% W -30% R CLOCKS
1691                                   ;&ALL MODULES
1692  446B  21  80  00         10.0  TST06:  LXI     H,128                  ;SET UP THE BYTE COUNT
1693  446E  22  78  45         16.0           SHLD    SIZE                   ;
1694                                                                        ;
1695  4471  3E  1B               7.0  TST06X: MVI     A,@033                 ;SET -20% W AND -30% R CLOCKS
1696  4473  32  06  43         13.0           STA     SPEED                  ;
1697  4476  CD  A9  44         18.0           CALL    PESUB                  ;CALL SUBROUTINE PESUB
1698  4479  2A  78  45         16.0           LHLD    SIZE                   ;GET THE BYTE COUNT
1699  447C  01  80  00         10.0           LXI     B,128                  ;SET UP THE INCREMENT VALUE
1700  447F  09                  10.0           DAD     B                      ;INCREMENT THE BYTE COUNT
1701  4480  22  78  45         16.0           SHLD    SIZE                   ;SAVE THE NEW BYTE COUNT
1702  4483  7C                   4.0           MOV     A,H                    ;DO UNTIL THE BYTE COUNT=
1703  4484  FE  03               7.0           CPI     $3
1704  4486  C2  71  44         10.0           JNZ     TST06X
1705  4489  21  58  04         10.0           LXI     H,1112.                ;SET UP THE BYTE COUNT
1706  448C  22  78  45         16.0           SHLD    SIZE                   ;
1707  448F  CD  A9  44         18.0           CALL    PESUB                  ;CALL SUBROUTINE PESUB
1708  4492                                   ENDTST  TST06
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4492                                   REQ     7                      ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4492  CD  06  28         18.0                      CALL    REQST
 (2)  4495  00                                           .BYTE                ;DATA PATTERN NUMBER
 (2)  4496  00  00                                       .WORD                ;SYSTEM "" COUNT
 (2)  4498  00  00                                       .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  449A  00                                           .BYTE                ;DATA COMPARE FLAG IF =1
 (2)  449B  07                                           .BYTE   7            ;REQUEST CODE
 (1)  449C  3A  9A  4F         13.0           LDA     ITERA                  ;GET ITERATION COUNT
 (1)  449F  3D                  4.0           DCR     A                      ;DOWNCOUNT
 (1)  44A0  32  9A  4F         13.0           STA     ITERA                  ;SAVE COUNT
 (1)  44A3  F2  6B  44         10.0           JP      TST06                  ;DO TEST UNTIL TILL = 0
1709  44A6  C3  18  28         10.0  EXIT:    JMP     TSTEND
```

```
1711                                        .SBTTL  SUBROUTINE PESUB
1712
1713  44A9  AF                    4.0  PESUB:  XRA     A               ;CLEAR THE ACCUMULATOR
1714  44AA  32  76  45           13.0         STA     PGCTL           ;SET THE PE FLAG
1715  44AD  C3  B5  44           10.0         JMP     PEGCOM
1716
1717  44B0  3E  FF                7.0  GCRSUB: MVI     A,@377          ;SET THE GCR FLAG
1718  44B2  32  76  45           13.0         STA     PGCTL
1719
1720  44B5  06  00                7.0  PEGCOM: MVI     B,@0
1721  44B7  3A  77  45           13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
1722  44BA  E6  01                7.0         ANI     @01             ;PORT 0?
1723  44BC  C2  D2  44           10.0         JNZ     FOUND           ;YES-GO USE PORT #0
1724  44BF  04                    4.0         INR     B               ;NO-UPDATE POINTER TO PORT #1
1725  44C0  3A  77  45           13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
1726  44C3  E6  02                7.0         ANI     @02             ;PORT 1?
1727  44C5  C2  D2  44           10.0         JNZ     FOUND           ;YES-GO USE PORT #1
1728  44C8  04                    4.0         INR     B               ;NO-UPDATE POINTER TO PORT #2
1729  44C9  3A  77  45           13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
1730  44CC  E6  04                7.0         ANI     @04             ;PORT 2?
1731  44CE  C2  D2  44           10.0         JNZ     FOUND           ;YES-GO USE PORT #2
1732  44D1  04                    4.0         INR     B               ;NO-ASSUME PORT #3
1733  44D2  CD  4B  45           18.0  FOUND:  CALL    CLEAR
1734  44D5  DB  E0               10.0         IN      INTSTA
1735  44D7  E6  80                7.0         ANI     BIT7
1736  44D9  B0                    4.0         ORA     B
1737  44DA  D3  E0               10.0         OUT     MBSEL           ;SELECT THE DESIRED TU-PORT AND MASSBUS
1738
1739  44DC  AF                    4.0  T1L01:  XRA     A               ;CLEAR THE DATA BUS CONTROL
1740  44DD  32  28  4F           13.0         STA     DBUS            ;MASK TO KEEP THE MASSBUS CLEAN
1741                                                                  ;
1742  44E0  2A  78  45           16.0         LHLD    SIZE            ;GET THE RECORD SIZE
1743  44E3  22  23  4F           16.0         SHLD    BYTEL           ;TO 128 BYTES
1744  44E6  7D                    4.0         MOV     A,L             ;GET THE BYTE COUNT LOW
1745  44E7                                    ROUT    RO5L            ;SAVE IN BYTE COUNT LOW
 (1)  44E7  D3  8A               10.0                 OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  44E9  7F                    4.0                 MOV     A,A             ;RETRY LINK
1746  44EA  7C                    4.0         MOV     A,H             ;GET THE BYTE COUNT HIGH
1747  44EB                                    ROUT    RO5H            ;SAVE IN BYTE COUNT HIGH
 (1)  44EB  D3  8B               10.0                 OUT     RO5H            ;WRITE AC INTO RO5H
 (1)  44ED  7F                    4.0                 MOV     A,A             ;RETRY LINK
1748                                                                  ;
1749  44EE  3A  7A  45           13.0         LDA     FORMT           ;GET THE FORMAT
1750  44F1  32  25  4F           13.0         STA     FORMAT          ;NO SKIP COUNT
1751  44F4                                    ROUT    RO2H            ;SAVE IN THE CAS
 (1)  44F4  D3  85               10.0                 OUT     RO2H            ;WRITE AC INTO RO2H
 (1)  44F6  7F                    4.0                 MOV     A,A             ;RETRY LINK
1752                                                                  ;
1753  44F7  3E  88                7.0         MVI     A,DDRCO         ;CLEAR DDR IN, SO DDR
1754  44F9  D3  DB               10.0         OUT     DDRCTL          ;WRITES TO TAPE
1755                                                                  ;
1756  44FB  2A  7B  45           16.0         LHLD    KWDAT           ;GET THE WRITE DATA
1757  44FE  29                   10.0         DAD     H               ;SHIFT IT LEFT 1 BIT POSITION
1758  44FF  17                    4.0         RAL                     ;SAVE THE CARRY
```

```
1759  4500  29              10.0          DAD     H               ;SHIFT IT LEFT ANOTHER BIT POSITION
1760  4501  17               4.0          RAL                     ;SAVE THAT CARRY TOO.
1761  4502  E6  03           7.0          ANI     $3              ;SAVE ONLY THE CARRY BITS
1762  4504  47               4.0          MOV     B,A             ;SAVE IN REGISTER B
1763  4505  7C               4.0          MOV     A,H             ;
1764  4506  D3  D9          10.0          OUT     DDRB            ;LOAD <13:6> IN DDR B
1765  4508  3A  7D  45      13.0          LDA     KWDAT+2         ;GET BITS <17:16>
1766  450B  85               4.0          ADD     L               ;MERGE WITH <5:0>
1767  450C  D3  D8          10.0          OUT     DDRA            ;LOAD INTO DDR A
1768  450E  AC               4.0          XRA     H               ;COMPUTE PARITY ON ALL BITS
1769  450F  A8               4.0          XRA     B               ;
1770  4510  78               4.0          MOV     A,B             ;GET BITS <15:14> AGAIN
1771  4511  E2  16  45       10.0          JPO     1$              ;JUMP IF ODD PARITY
1772  4514  F6  04           7.0          ORI     BIT2            ;SET THE PARITY BIT TO MAKE IT ODD
1773  4516  D3  DA          10.0  1$:     OUT     DDRC            ;LOAD <P:14> IN DDRC
1774                                                              ;
1775  4518  21  03  43      10.0          LXI     H,SPEED-3       ;POINT TO THE CLOCK
1776  451B  22  26  4F      16.0          SHLD    CLOCK           ;SPEED TABLE
1777  451E  3A  76  45      13.0          LDA     PGCTL           ;GET THE PE/GCR FLAG
1778  4521  A7               4.0          ANA     A               ;SET CONDITION BITS
1779  4522  C2  2B  45      10.0          JNZ     2$              ;GO DO GCR
1780  4525  CD  4F  00      18.0          CALL    LKLWPP          ;ELSE-DO PE
1781  4528  C3  2E  45      10.0          JMP     3$              ;
1782  452B  CD  52  00      18.0  2$:     CALL    LKLWPG          ;CALL LOOP WRITE/READ - TU PORT - GCR
1783                                                              ;
1784  452E  D2  44  45      10.0  3$:     JNC     T1CN1                   ;CONTINUE IF NO ERROR
1785  4531  78               4.0          MOV     A,B                     ;GET THE FAILURE CODE
1786  4532  E6  FC           7.0          ANI     $FC                     ;MASK OUT THE MEANINGFUL BITS
1787  4534  0F               4.0          RRC                             ;POSITION FOR OUTPUT
1788  4535  0F               4.0          RRC                             ;
1789  4536                               ROUT    ADATA                   ;STORE IN THE CAS
 (1)  4536  D3  94           10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4538  7F               4.0                   MOV     A,A             ;RETRY LINK
1790  4539  79               4.0          MOV     A,C                     ;GET THE INTERRUPT CODE
1791  453A  E6  3F           7.0          ANI     $3F                     ;MASK OUT THE MEANINGFUL BITS
1792  453C                               ROUT    EDATA                   ;STORE IN THE CAS
 (1)  453C  D3  95           10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  453E  7F               4.0                   MOV     A,A             ;RETRY LINK
1793  453F                               ERRB    T1LO1,T1CN1,3
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA 'EDATA'' VALID
 (1)  453F  CD  12  28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0002                                   MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4542  02                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4543  03                                     .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4544  CD  15  28       18.0          T1CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4547  DA  DC  44       10.0                  JC      T1LO1           ;LOOP ADDRESS IF LOOP SPECIFIED
1794                                   ;>LOOP WRITE/READ ERROR
1795                                   ;<ACTUAL DATA IS THE FAILURE CODE
1796                                   ;<EXPECTED DATA IS THE INTERRUPT CODE
1797  454A  C9                10.0          RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TL PORTS
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************

                        12.0  CLEAR:  PUSH    PSW           ;SAVE THE SELECTED PORT #
                        12.0          PUSH    B
                         7.0          MVI     B,0           ;START TO CLEAR AT PORT #0
                        10.0  CLRLP:  IN      INTSTA        ;GET MB SELECT INFO
                         7.0          ANI     BIT7          ;SAVE ONLY THE MASSBUS SELECT BIT
                         4.0          ORA     B             ;ADD IN THE SELECTED PORT #
                        10.0          OUT     MBSEL         ;RESET TO THIS PORT
                         7.0          MVI     A,@200        ;LOAD MTA REGISTER #0 SELECT CODE
                        10.0          OUT     TCMD          ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                         4.0          XRA     A             ;CLEAR TU COMMAND A
                        10.0          OUT     TCMD          ;
                         7.0          MVI     A,@201        ;LOAD MTA REGISTER #1 SELECT CODE
                        10.0          OUT     TCMD          ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                         7.0          MVI     A,SELCLR      ;LOAD TU "CLEAR SELECT" COMMAND
                        10.0          OUT     TCMD          ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
                         4.0          XRA     A
                        10.0          OUT     TAMT          ;CLEAR AMTIE WORD
                        10.0          OUT     PDIAG         ;CLEAR DIAG CONTROL WORD
                        10.0          OUT     PENAB         ;CLEAR PORT ENABLE WORD
                         4.0          INR     B             ;POINT TO THE NEXT PORT TO CLEAR
                         4.0          MOV     A,B
                         7.0          CPI     4             ;DONE?
                        10.0          JNZ     CLRLP         ;NO - CLEAR THIS PORT ALSO
                        10.0          POP     B             ;RESET B & C
                        10.0          POP     PSW           ;ALL DONE
                        10.0          RET                   ;EXIT
```

```
1844
1845
1846  4576   00
1847  4577   00
1848  4578   00    00
1849  457A   00
1850  457B   00
1851  457C   00
1852  457D   00
1853         0000

.SBTTL  PROGRAM VARIABLES

PGCTL:  .BYTE   0               ;PE/GCR CONTROL FLAG
UNITMP: .BYTE   0               ;UNIT MAP
SIZE:   .WORD   0               ;RECORD SIZE
FORMT:  .BYTE   0               ;DATA FORMAT
KWDAT:  .BYTE   0               ;18 BITS OF INPUT DATA
        .BYTE   0
        .BYTE   0
        .END
```

```
A      =%0007      ADATA  = 0094      AMTIEP= 0001      AMTIE7= 0002
ARAIDF= 0098       ASAVE    4F9B      ATTCD    4F97     AXNUM    4F91
B      =%0000      BADST  = 0090      BIT0   = 0001      BIT1   = 0002
BIT15  = 8000      BIT2   = 0004      BIT3   = 0008      BIT4   = 0010
BIT5   = 0020      BIT6   = 0040      BIT7   = 0080      BIT8   = 0100
BIT9   = 0200      BRKPBC= 4F0A       BRKRAM= 4F10      BRKSTR= 4E60
BRKXCT= 4F00       BSAVE    4F9C      BYTCNT= 00D4      BYTEH    4F24
BYTEL    4F23      C      =%0001      CASCT    4F21     CASCTL= 00A0
CASSTA= 00A0       CATTH  = 0089      CAT1_  = 0088      CBUSST= 00A1
CBYTH  = 0088      CBYTL  = 008A      CDG1H  = 0087      CDG1L  = 0086
CDG2H  = 0093      CDG2L  = 0092      CDG3H  = 0095      CDG3L  = 0094
CDVTH  = 008D      CDVTL  = 008C      CH0TIE= 0028       CH0TIE= 0020
CH1TIE= 0021       CH2TIE= 0022       CH3TIE= 0023       CH4TIE= 0024
CH5TIE= 0025       CH6TIE= 0026       CH7TIE= 0027       CKLOP  = 2815
CLEAR    454B      CLKCTL= 00F0       CLOCK    4F26      CLRLP    454F
CMC0H  = 0099      CMC0L  = 0098      CMC1H  = 009B      CMC1L  = 009A
CMC2H  = 009D      CMC2L  = 009C      CMC3H  = 009F      CMC3L  = 009E
CMINH  = 0097      CMINL  = 0096      CNTCTL= 00D7       CRCWRD= 0018
CSAVE    4F9D      CSRLH  = 0091      CSRLL  = 0090      CTCH   = 0085
CTCL   = 0084      CTSTH  = 008F      CTSTL  = 008E      CXCTH  = 0081
CXCTL  = 0080      CXINH  = 0083      CXINL  = 0082      C.     = 0001
C.AVAI= 0080       C.DP   = 0008      C.DSE  = 0010      C.DTU  = 0003
C.DVA  = 0008      C.FAIL= 00FC       C.FMT  = 0070      C.FNCT= 003E
C.GO   = 0001      C.INTC= 00FE       C.MAIN= 0020       C.NSA  = 0080
C.RCT  = 00FC      C.SER  = 0080      C.SHR  = 0040      C.SKPC= 000F
C.TAPE= 0040       C.WCS  = 0002      D      =%0002      DATACT= 00D0
DBUS     4F28      DBUSCT= 00C0       DBUSST= 00C0       DDRA   = 00D8
DDRAIN= 0010       DDRB   = 00D9      DDRBIN= 0002       DDRC   = 00DA
DDRCIN= 0001       DDRCO  = 0088      DDRCTL= 00DB       DIAFLG   4F22
DIAGPG= 4300       DIAGRM= 4F90       DIARD  = 000B      DONE1  = 0045
DONINT= 0010       DSAVE    4F9E      DSE    = 0006      DUMMY    4325  G
D.ATH0= 0001       D.ATH1= 0002       D.EOTD= 0010       D.LAGC= 0020
D.NOTW= 0040       D.NTHR= 0004       D.TACH= 0008       D.WR4  = 0080
E      -%0003      ECCBAD= 0042       ECCCOR= 0019       ECCOK  = 0041
ECCSTA= 001A       ECCTST= 000E       EDATA  = 0095      EOTCLR= 0003
ERFLG    4F93      ERLP   = 2809      ERLPA  = 280F      ERLPB  = 2812
ERLPE  = 280C      ERNUM    4F90      ERRCNT= 00D6       ESAVE    4F9F
EXIT     44A6      E.ACRC= 0010       E.AMT  = 0020      E.CDP  = 0080
E.CRC  = 0080      E.PNTR= 0008       E.RPE  = 0040      E.STEC= 0001
E.TTEC= 0002       E.UNC  = 0004      FIFORD= 006A       FORMAT   4725
FORMT    457A      FOUND    44D2      FWDTST= 0061       GCRID  = 0089
GCRSET= 0002       GCRSUB   44B0      GOODTM= 0092       H      =%0004
HLSAVE   4FA0      IE     = 0008      INTSTA= 00E0       ITERA    4F9A
I.PWR  = 0020      I.RMPE= 0040       I5.5   = 0010      I6.5   = 0020
I7.5   = 0040      KCALL  = 005F      KCLR   = 007B      KDEP   = 003F
KENAB  = 0078      KEXAM  = 003E      KEYBRD= 00C8       KEY1   = 0078
KEY10  = 006D      KEY11  = 006E      KEY12  = 006F      KEY13  = 005C
KEY14  = 005D      KEY15  = 005E      KEY16  = 005F      KEY17  = 003C
KEY18  = 003D      KEY19  = 003E      KEY2   = 0079      KEY20  = 003F
KEY3   = 007A      KEY4   = 007B      KEY5   = 0074      KEY6   = 0075
KEY7   = 0076      KEY8   = 0077      KEY9   = 006C      KINTA  = 006F
KLDAD  = 003D      KN0    = 003C      KN1    = 005C      KN2    = 005D
KN3    = 005E      KN4    = 006C      KN5    = 006D      KN6    = 006E
KN7      0074      KN8    = 0075      KN9    = 0076      KU2    = 0079
```

```
KU3    = 007A          KU8    = 0077          KWDAT   4578          L      =%0005
.BLANK= 000F          LCE    = 0008          LCH    = 000C          LCL    = 000D
.CP    = 000E          LC0    = 0000          LC1    = 0001          LC2    = 0002
LC3    = 0003          LC4    = 0004          LC5    = 0005          LC6    = 0006
LC7    = 0007          LC8    = 0008          LC9    = 0009          LDLEDA= 00CA
LDLEDB= 00CB          LDLEDC= 00CC          LDLEDD= 00CD          LDLEDE= 00CE
LDLEDF= 00CF          LKDIAG= 2800          LKKBD  = 004C          LKKEY  = 0049
LKLWMG= 0058          LKLWMP= 0055          LKLWPG= 0052          LKLWPP= 004F
LKMOD7= 0046          LKOPR  = 0046          LPFLG   4F94          LPNUM   4F92
M      =%0006          MBSEL  = 00E0          MB.A   = 0008          MB.B   = 0004
MEMTOP= 4FFF          MINUS  = 000A          MM     = 8000          MSE    = 0008
MSGN   = 0002          MTACLR= 0000          MT.ARA= 0020          MT.CPE= 0080
MT.DSE= 0001          MT.FWD= 0040          MT.INH= 0008          MT.LWR= 0004
MT.MOT= 0002          MT.NWT= 0080          MT.PEC= 0040          MT.PSB= 0004
MT.PSO= 0001          MT.PS  = 0002          MT.REV= 0020          MT.WRT= 0010
MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020          M.CONT= 0080
M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008          M.FAIL= 0008
M.ILR  = 0010          M.INIT= 0010          M.OCC  = 0020          M.ONLI= 0001
M.PE   = 0040          M.PORT= 0080          M.RDEN= 0002          M.RDPE= 0008
M.RUN  = 0004          M.SCLK= 0001          M.TRA  = 0040          M.UNIT= 0007
M.WCLK= 0040          M.WCLN= 0080          M.WREN= 0080          M5.5   = 0001
M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088          OKAY   = 00FF
OPRRAM= 4300          OPSTRT= 0058          OPVER  = 0040          PADCNT= 00D5
PADCRC= 0080          PDIAG  = 0048          PEGCOM  44B5          PEID   = C08A
PENAB  = 004C          PESET  = 0001          PESUB   44A9          PGCTL   4576
PL     = 0081          PRDD   = 004C          PRENF  = 009C          PS     = 0082
PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001          P.BCTC= 0040
P.CMDP= 0020          P.INTE= 0080          P.LCS  = 0040          P.LWR  = 0020
P.RDP  - 0002          P.RPEN= 0004          P.RPST= 0002          P.RP0E= 0020
P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010          P.SING= 0080
P.STAT= 0002          P.STPE= 0080          P.TACH= 0008          P.TUPR= 0010
P.WCSP= 0004          P.WDS  = 0040          P.WFLP= 0001          P.WPEN= 0010
P.WP0E= 0008          P.WP1E= 0004          P.WP2E= 0008          P.WP3E= 0004
P.5VOK= 0002          QUE    = 281B          QUEM   = 281E          RAMT   = 0010
RARA   = 0006          RARAI  = 0004          RCHBD0= 0048          RCHBD1= 0047
RCHOK  = 0046          RCHTST= 000C          RCLRT  = 000D          RCMD   = 0008
RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017          RDCLK  = 0010
RDON   = 0011          READG  = 0007          REND   = 0014          REQST  = 2806
RESCHR= 00D1          REVTST= 0064          REWIND= 0004          RFIFOL= 0008
RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001          RILL   = 0012
RINST  = 000C          RMCTST= 0008          RMK2   = 0013          RNOP   = 0000
RPATH  - 0001          RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003
RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000          RPF1   = 009D
RPF2   = 009E          RPOK   = 0043          RPOSIN= 0016          RPSTA  = 0015
RRCMT  = 000A          RSTAT  = 0002          RTIcB  = 000A          RTIER  = 0030
RTM    = 0005          RUNKI  = 0009          RUPTST= 005E          RWDUNL= 0005
R.AMT  = 0001          R.BOP  = 0008          F.DATA= 0040          R.DON  = 0002
R.DRDY= 0010          R.END  = 0010          R.ILL  = 0004          R.JVOK= 0004
R.MK2  = 0008          R.PLOD= 0008          R.PLO0= 0010          R.PLO1= 0020
R.POST= 0020          R.STNM= 0002          R.STOP= 0004          R.STPC= 0001
R.TBJN= 0080          R.TSTD= 0040          R.VOK  = 0080          R00H   = 0081
R00L   = 0080          R01H   = 0083          R01L   = 0082          R02H   = 0085
R02L   = 0084          R03H   = 0087          R03L   = 0086          R04H   = 0089
R04L   = 0088          R05H   = 0088          R05L   = 008A          R06H   = 008D
```

TUP2.M80 SYMBOL TABLE

```
R06L  = 008C        R07H  = 008F        R07L  = 008E        R10H  = 0091
R10L  = 0090        R11H  = 0093        R11L  = 0092        R12H  = 0095
R12L  = 0094        R13H  = 0097        R13L  = 0096        R14H  = 0099
R14L  = 0098        R15H  = 009B        R15L  = 009A        R16H  = 009D
R16L  = 009C        R17H  = 009F        R17L  = 009E        R7.5  = 0010
SELCLR= 0000        SETATA= 00A1        SID   = 0080        SIZE    4578
SOD   = 0080        SOE   = 0040        SP    =%0008        SPEED   4306
SSCLK = 0040        SSTEP = 0005        STACK = 4FFF        STATRM= 4F20
STPCT   4F20        STRSP = 5000        TADR00= 0080        TADR01= 0081
TADR02= 0082        TADR03= 0083        TADR04= 0084        TADR05= 0085
TADR06= 0086        TADR07= 0087        TADR10= 0088        TADR11= 0089
TADR12= 008A        TADR13= 008B        TAMT  = 0044        TASEL = 0080
TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004
TC.REV= 0020        TC.WRT= 0010        TEMP    4F99        TEST02  4366
TEST03  43A6        TEST04  43E6        TEST05  4426        TEST06  4466
TEST1   4307        TMF   = 0099        TMRDY = 0040        TRKENA= 00D2
TSET  = 2803        TSTEND= 2818        TSTS  = 0040        TST01X  432B
TST01Y  4331        TST02   436B        TST02X  4371        TST03   43AB
TST03X  43B1        TST04   43EE        TST04X  43F1        TST05   442B
TST05X  4431        TST06   446B        TST06X  4471        TUSEL0= 00D1
TUSEL1= 00D2        TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002
T.BOT = 0004        T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004
T.ONL = 0020        T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008
T.PS1J= 0010        T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010
T.SCLK= 0002        T1CN1   4544  G     T1LO1   44DC        UIBG  = 00A1
UNITMP  4577        VALFC   4F98        VALTB   4F95        VELTST= 005B
WDR.P = 0010        WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0
WRTCLK= 0000        WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008
W.DIAG= 0002        W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080
W.ERR = 0020        W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004
W.ONES= 0020        W.RESI= 0002        W.REV = 0004        W.ROME= 0010
W.RST = 0001        W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020
X     =%000A        X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002
X.ROME= 0001        X.WCLK= 0001        Y     =%000B        .     = 457E
```

ERRORS DETECTED: 0

*TUP2.A78/PTP,TUP2=NLIST,PARAM,MACRO,LIST,TUP2
RUN-TIME: 3 4 0 SECONDS
CORE USED: 10K

```
                                .TITLE  MTA4 - MAG TAPE ADAPTER TEST PART #4
                                ;ID     MTA4-MAG TAPE ADAPTER CONTROLLER PART #4
                                .SBTTL  TEST 1 - LOOP WRITE/READ - MTA - GCR - NORMAL W/R CLOCKS
1332  4300                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1333                            ;*GCR - LOOP WRITE/READ - MTA - NORMAL W/R CLOCKS
1334  4300                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1335                            ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
1336                            ;*GCR FORMAT WITH NORMAL WRITE AND READ CLOCKS
1337                            ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
1338  4300                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1339                            ;*BGNTST
1340                            ;*  GET USER SELECTED TU PORT (MTA)
1341                            ;*  SET THE RECORD SIZE TO 128
1342                            ;*  BGNDO
1343                            ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
1344                            ;*  :  SET THE READ PATH CLOCK TO NORMAL
1345                            ;*  :  CALL SUBROUTINE GCRSUB
1346                            ;*  :  ADD 128 TO THE RECORD SIZE
1347                            ;*  :  DO UNTIL THE RECORD SIZE 640
1348                            ;*  ENDDO
1349                            ;*  SET RECORD SIZE TO 1112.
1350                            ;*  CALL GCRSUB SUBROUTINE
1351                            ;*ENDTST
1352  4300                      SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1353                            ;*MTA4 MICRO TEST 01
1354                            ;*MTA4 MICRO ERROR 01
1355                            ;*MTA4-LOOP WRITE/READ - MTA - GCR - NORMAL W/R CLOCKS
1356                            ;*ALL MODULES
1357                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1358                            ;*FATAL ERROR - TEST ABORTED
1359                            ;*
1360                            ;*MTA4 MICRO TEST 01
1361                            ;*MTA4 MICRO ERROR 02
1362                            ;*MTA4-LOOP WRITE/READ - MTA - GCR - NORMAL W/R CLOCKS
1363                            ;*ALL MODULES
1364                            ;*LOOP WRITE/READ ERROR
1365                            ;*ACTUAL = AAAA
1366                            ;*EXPECTED = EEEE
1367                            ;*BYTE-SCLK COUNT = LLL
1368                            ;*DATA FORMAT = MM
1369                            ;*SKIP COUNT - NN
1370                            ;*ACTUAL DATA IS THE FAILURE CODE
```

```
1371                                    ;*EXPECTED DATA IS THE INTERRUPT CODE
1372  4300                              $
 (1)                                    ; ***************************************************************************
1373
1374  4300  C3  07  43      10.0         JMP     TEST1
1375  4303  00                          .BYTE   0               ;DUMMY BYTE
1376  4304  00                          .BYTE   0               ;DUMMY BYTE
1377  4305  00                          .BYTE   0               ;DUMMY BYTE
1378  4306  00                SPEED:    .BYTE   0               ;CLOCK CONTROL WORD
1379  4307                    TEST1:    TESTX   @1              ;INITIALIZE THE TEST
 (1)  4307  3E  01               7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4309  CD  03  28         18.0                  CALL    TSET            ;SETUP THE TEST
1380                                    ;%MTA4-LOOP WRITE/READ - MTA - GCR - NORMAL W/R CLOCKS
1381                                    ;&ALL MODULES
1382  430C                              REQ     @7,0,0,0,0
 (1)  430C  CD  06  28         18.0                  CALL    REQST
 (1)  430F  00                                       .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4310  00  00                                   .WORD   0               ;SYSTEM "0" COUNT
 (1)  4312  00  00                                   .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4314  00                                       .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  4315  07                                       .BYTE   @7              ;REQUEST CODE
1383  4316                              RIN     R12L
 (1)  4316  DB  94              10.0                 IN      R12L            ;READ R12L INTO AC
 (1)  4318  7F                   4.0                 MOV     A,A             ;RETRY LINK
1384  4319  32  3F  46          13.0         STA     UNITMP
1385  431C  A7                    4.0         ANA     A
1386  431D  C2  28  43          10.0         JNZ     TST01X
1387  4320                               ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4320  CD  09  28         18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4323  01                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4324  00                                       .BYTE
 (1)  4325  CD  15  28         18.0          DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4328  DA  62  45         10.0                  JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1388                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1389                                    ;>FATAL ERROR - TEST ABORTED
```

```
1391  432B  21  80  00      10.0  TST01X: LXI   H,128              ;SET UP THE BYTE COUNT
1392  432E  22  40  46      16.0          SHLD  SIZE               ;
1393                                                              ;
1394  4331  3E  10           7.0  TST01Y: MVI   A,@020             ;SET UP FOR NORMAL CLOCKS
1395  4333  32  06  43      13.0          STA   SPEED
1396  4336  CD  6C  45      18.0          CALL  GCRSUB         ;CALL SUBROUTINE LWRSUB
1397                                                              ;
1398  4339  2A  40  46      16.0          LHLD  SIZE               ;GET THE SIZE
1399  433C  01  80  00      10.0          LXI   B,128              ;LOAD THE INCRFMENT VALUE
1400  433F  09              10.0          DAD   B                  ;INCREMENT THE BYTE COUNT
1401  4340  22  40  46      16.0          SHLD  SIZE               ;SAVE THE SIZE FOR LATER
1402  4343  7C               4.0          MOV   A,H                ;DO UNTIL THE BYTE COUNT = 768
1403  4344  FE  03           7.0          CPI   $3                 ;
1404  4346  C2  31  43      10.0          JNZ   TST01Y             ;
1405  4349  21  58  04      10.0          LXI   H,1112.            ;SET UP THE BYTE COUNT
1406  434C  22  40  46      16.0          SHLD  SIZE               ;
1407  434F  CD  6C  45      18.0          CALL  GCRSUB             ;CALL SUBROUTINE GCRSUB
1408  4352                                ENDTST TST01X
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4352                              REQ   7                  ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4352  CD  06  28      18.0                           CALL    REQST
 (2)  4355  00                                             .BYTE                ;DATA PATTERN NUMBER
 (2)  4356  00  00                                         .WORD                ;SYSTEM '*' COUNT
 (2)  4358  00  00                                         .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  435A  00                                             .BYTE                ;DATA COMPARE FLAG IF =1
 (2)  435B  07                                             .BYTE   7            ;REQUEST CODE
 (1)  435C  3A  9A  4F      13.0          LDA   ITERA              ;GET ITERATION COUNT
 (1)  435F  3D               4.0          DCR   A                  ;DOWNCOUNT
 (1)  4360  32  9A  4F      13.0          STA   ITERA              ;SAVE COUNT
 (1)  4363  F2  28  43      10.0          JP    TST01X             ;DO TEST UNTIL TILL - 0
1409
```

```
                                .SBTTL  TEST 2 - LOOP WRITE/READ - MTA - GCR - +10% W/R CLOCKS

                                ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR - LOOP WRITE/READ - MTA +10% W/R CLOCKS
                                SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE-640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE GCRSUB
                                ;*ENDTST
                                SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA4 MICRO TEST 02
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - GCR - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT - LLL
                                ;*DATA FORMAT - MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ;*****************************************************************************
```

```
1449  4366                                  TEST02: TESTX   @2               ;INITIALIZE THE TEST
 (1)  4366  3E 02              7.0                        MVI     A,@2        ;DEFINE THE TEST NUMBER
 (1)  4368  CD 03 28          18.0                        CALL    TSET        ;SETUP THE TEST
1450                                   ;%MTA4-LOOP WRITE/READ - MTA - GCR - +10% W/R CLOCKS
1451                                   ;&ALL MODULES
1452  436B  21 80 00          10.0    TST02:  LXI     H,128               ;SETUP THE BYTE COUNT
1453  436E  22 40 46          16.0            SHLD    SIZE                ;
1454                                                                       ;
1455  4371  3E 0A               7.0    TST02X: MVI     A,@012              ;SET W/R CLOCKS TO +10%
1456  4373  32 06 43           13.0            STA     SPEED               ;
1457  4376  CD 6C 45           18.0            CALL    GCRSUB              ;CALL SUBROUTINE LWRSUB
1458  4379  2A 40 46           16.0            LHLD    SIZE                ;GET THE BYTE COUNT
1459  437C  01 80 00           10.0            LXI     B,128               ;SET UP INCREMENT VALUE
1460  437F  09                 10.0            DAD     B                   ;INCREMENT THE BYTE COUNT
1461  4380  22 40 46           16.0            SHLD    SIZE                ;SAVE THE NEW BYTE COUNT
1462  4383  7C                  4.0            MOV     A,H                 ;DO UNTIL THE BYTE COUNT=1
1463  4384  FE 03               7.0            CPI     $3
1464  4386  C2 71 43           10.0            JNZ     TST02X
1465  4389  21 58 04           10.0            LXI     H,1112.                     ;SET UP THE BYTE COUNT
1466  438C  22 40 46           16.0            SHLD    SIZE                        ;
1467  438F  CD 6C 45           18.0            CALL    GCRSUB                      ;CALL SUBROUTINE GCRSUB
1468  4392                                    ENDTST  TST02
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4392                                      REQ     7                   ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4392  CD 06 28          18.0                                CALL    REQST
 (2)  4395  00                                                    .BYTE               ;DATA PATTERN NUMBER
 (2)  4396  00 00                                                 .WORD               ;SYSTEM '*' COUNT
 (2)  4398  00 00                                                 .WORD               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  439A  00                                                    .BYTE               ;DATA COMPARE FLAG IF =1
 (2)  439B  07                                                    .BYTE   7           ;REQUEST CODE
 (1)  439C  3A 9A 4F          13.0                LDA     ITERA               ;GET ITERATION COUNT
 (1)  439F  3D                 4.0                DCR     A                   ;DOWNCOUNT
 (1)  43A0  32 9A 4F          13.0                STA     ITERA               ;SAVE COUNT
 (1)  43A3  F2 6B 43          10.0                JP      TST02               ;DO TEST UNTIL TILL = 0
1469
```

```
                                .SBTTL  TEST 3 - LOOP WRITE/READ - MTA - GCR - -20% W -30% R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR - LOOP WRITE/READ - MTA - -20%W -30%R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO -20%
                                ;*  :  SET THE READ PATH CLOCK TO -30%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL GCRSUB SUBROUTINE
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA4 MICRO TEST 03
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - GCR - -20% W -30% R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************
```

```
1509  43A6                                  TEST03: TESTX   @3               ;INITIALIZE THE TEST
 (1)  43A6  3E  03              7.0                      MVI     A,@3        ;DEFINE THE TEST NUMBER
 (1)  43A8  CD  03  28         18.0                      CALL    TSET        ;SETUP THE TEST
1510                                   ;%MTA4-LOOP WRITE/READ - MTA - GCR - -20% W -30% R CLOCKS
1511                                   ;&ALL MODULES
1512  43AB  21  80  00         10.0    TST03:  LXI     H,128               ;SET UP THE BYTE COUNT
1513  43AE  22  40  46         16.0            SHLD    SIZE                ;
1514                                                                       ;
1515  43B1  3E  18               7.0    TST03X: MVI     A,@033              ;SET -20% W AND -30% R CLOCKS
1516  43B3  32  06  43          13.0            STA     SPEED               ;
1517  43B6  CD  6C  45          18.0            CALL    GCRSUB              ;CALL SUBROUTINE LWRSUB
1518  43B9  2A  40  46          16.0            LHLD    SIZE                ;GET THE BYTE COUNT
1519  43BC  01  80  00          10.0            LXI     B,128               ;SET UP THE INCREMENT VALUE
1520  43BF  09                  10.0            DAD     B                   ;INCREMENT THE BYTE COUNT
1521  43C0  22  40  46          16.0            SHLD    SIZE                ;SAVE THE NEW BYTE COUNT
1522  43C3  7C                   4.0            MOV     A,H                 ;DO UNTIL THE BYTE COUNT=
1523  43C4  FE  03               7.0            CPI     $3
1524  43C6  C2  B1  43          10.0            JNZ     TST03X
1525  43C9  21  58  04          10.0            LXI     H,1112.                     ;SET UP THE BYTE COUNT
1526  43CC  22  40  46          16.0            SHLD    SIZE                        ;
1527  43CF  CD  6C  45          18.0            CALL    GCRSUB                      ;CALL SUBROUTINE GCRSUB
1528  43D2                                    ENDTST  TST03
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43D2                                      REQ     7                   ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43D2  CD  06  28         18.0                              CALL    REQST
 (2)  43D5  00                                                   .BYTE                   ;DATA PATTERN NUMBER
 (2)  43D6  00  00                                               .WORD                   ;SYSTEM "" COUNT
 (2)  43D8  00  00                                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43DA  00                                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43DB  07                                                   .BYTE   7               ;REQUEST CODE
 (1)  43DC  3A  9A  4F         13.0                      LDA     ITERA               ;GET ITERATION COUNT
 (1)  43DF  3D                  4.0                      DCR     A                   ;DOWNCOUNT
 (1)  43E0  32  9A  4F         13.0                      STA     ITERA               ;SAVE COUNT
 (1)  43E3  F2  AB  43         10.0                      JP      TST03               ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 4 - LOOP WRITE/READ - MTA - PE - NORMAL W/R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE/READ - MTA - NORMAL W/R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH NORMAL WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
                                ;*  :  SET THE READ PATH CLOCK TO NORMAL
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL PESUB SUBROUTINE
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA4 MICRO TEST 04
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - PE - NORMAL W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************
```

```
1568  43E6                                  TEST04: TESTX   @4               ;INITIALIZE THE TEST
 (1)  43E6  3E  04              7.0                      MVI     A,@4         ;DEFINE THE TEST NUMBER
 (1)  43E8  CD  03  28         18.0                      CALL    TSET         ;SETUP THE TEST
1569                                   ;%MTA4-LOOP WRITE/READ - MTA - PE - NORMAL W/R CLOCKS
1570                                   ;&ALL MODULES
1571  43EB  21  80  00         10.0    TST04:  LXI     H,128                ;SET UP THE BYTE COUNT
1572  43EE  22  40  46         16.0            SHLD    SIZE                 ;
1573                                                                        ;
1574  43F1  3E  10               7.0    TST04X: MVI     A,@020               ;SET UP FOR NORMAL CLOCKS
1575  43F3  32  06  43          13.0            STA     SPEED
1576  43F6  CD  65  45          18.0            CALL    PESUB           ;CALL SUBROUTINE PESUB
1577                                                                        ;
1578  43F9  2A  40  46          16.0            LHLD    SIZE                 ;GET THE SIZE
1579  43FC  01  80  00          10.0            LXI     B,128                ;LOAD THE INCREMENT VALUE
1580  43FF  09                  10.0            DAD     B                    ;INCREMENT THE BYTE COUNT
1581  4400  22  40  46          16.0            SHLD    SIZE                 ;SAVE THE SIZE FOR LATER
1582  4403  7C                   4.0            MOV     A,H                  ;DO UNTIL THE BYTE COUNT - 768
1583  4404  FE  03               7.0            CPI     $3                   ;
1584  4406  C2  F1  43          10.0            JNZ     TST04X               ;
1585  4409  21  58  04          10.0            LXI     H,1112.              ;SET UP THE BYTE COUNT
1586  440C  22  40  46          16.0            SHLD    SIZE                 ;
1587  440F  CD  65  45          18.0            CALL    PESUB                ;CALL SUBROUTINE PESUB
1588  4412                                    ENDTST  TST04
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4412                                      REQ     7                    ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4412  CD  06  28         18.0                                  CALL    REQST
 (2)  4415  00                                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  4416  00  00                                                   .WORD                   ;SYSTEM '*' COUNT
 (2)  4418  00  00                                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  441A  00                                                       .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  441B  07                                                       .BYTE   7               ;REQUEST CODE
 (1)  441C  3A  9A  4F          13.0                    LDA     ITERA        ;GET ITERATION COUNT
 (1)  441F  3D                   4.0                    DCR     A            ;DOWNCOUNT
 (1)  4420  32  9A  4F          13.0                    STA     ITERA        ;SAVE COUNT
 (1)  4423  F2  EB  43          10.0                    JP      TST04        ;DO TEST UNTIL TILL = 0
1589
```

```
                                .SBTTL  TEST 5 - LOOP WRITE/READ - MTA - PE - +10% W/R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE/READ - MTA +10% W/R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL PESUB SUBROUTINE
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA4 MICRO TEST 05
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - PE - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************
```

```
1629  4426                                  TEST05: TESTX   @5               ;INITIALIZE THE TEST
 (1)  4426  3E  05              7.0                    MVI     A,@5            ;DEFINE THE TEST NUMBER
 (1)  4428  CD  03  28         18.0                    CALL    TSET            ;SETUP THE TEST
1630                                   ;%MTA4-LOOP WRITE/READ - MTA - PE - +10% W/R CLOCKS
1631                                   ;&ALL MODULES
1632  442B  21  80  00         10.0    TST05:  LXI     H,128           ;SETUP THE BYTE COUNT
1633  442E  22  40  46         16.0            SHLD    SIZE            ;
1634                                                                   ;
1635  4431  3E  0A              7.0    TST05X: MVI     A,@012          ;SET W/R CLOCKS TO +10%
1636  4433  32  06  43         13.0            STA     SPEED           ;
1637  4436  CD  65  45         18.0            CALL    PESUB           ;CALL SUBROUTINE PESUB
1638  4439  2A  40  46.        16.0            LHLD    SIZE            ;GET THE BYTE COUNT
1639  443C  01  80  00'        10.0            LXI     B,128           ;SET UP INCREMENT VALUE
1640  443F  09                  10.0            DAD     B               ;INCREMENT THE BYTE COUNT
1641  4440  22  40  46         16.0            SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1642  4443  7C                   4.0            MOV     A,H             ;DO UNTIL THE BYTE COUNT=1
1643  4444  FE  03              7.0            CPI     $3
1644  4446  C2  31  44         10.0            JNZ     TST05X
1645  4449  21  58  04         10.0            LXI     H,1112.                 ;SET UP BYTE COUNT
1646  444C  22  40  46         16.0            SHLD    SIZE                    ;
1647  444F  CD  65  45         18.0            CALL    PESUB                   ;CALL SUBROUTINE PESUB
1648  4452                                    ENDTST  TST05
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4452   .                                  REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4452  CD  06  28         18.0                                    CALL    REQST
 (2)  4455  00                                                         .BYTE                   ;DATA PATTERN NUMBER
 (2)  4456  00  00                                                     .WORD                   ;SYSTEM '*' COUNT
 (2)  4458  00  00                                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  445A  00                                                         .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  445B  07                                                         .BYTE   7               ;REQUEST CODE
 (1)  445C  3A  9A  4F         13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  445F  3D                  4.0            DCR     A               ;DOWNCOUNT
 (1)  4460  32  9A  4F         13.0            STA     ITERA           ;SAVE COUNT
 (1)  4463  F2  2B  44         10.0            JP      TST05           ;DO TEST UNTIL TILL = 0
1649
```

```
                                .SBTTL  TEST 6 - LOOP WRITE/READ - MTA - PE - -20% W -30% R CLOCKS

                                ST
                                ; ******************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE/READ - MTA - -20%W -30%R CLOCKS
                                SD
                                ; ******************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ******************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO -20%
                                ;*  :  SET THE READ PATH CLOCK TO -30%
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL PESUB SUBROUTINE
                                ;*ENDTST
                                SE
                                ; ******************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA4 MICRO TEST 06
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - PE - -20% W -30% R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ******************************************************************************
```

```
1689  4466                                  TEST06: TESTX   @6               ;INITIALIZE THE TEST
 (1)  4466  3E 06              7.0                       MVI     A,@6         ,DEFINE THE TEST NUMBER
 (1)  4468  CD 03 28          18.0                       CALL    TSET         ;SETUP THE TEST
1690                                   ;%MTA4-LOOP WRITE/READ - MTA - PE - -20% W -30% R CLOCKS
1691                                   ;&ALL MODULES
1692  446B  21 80 00          10.0    TST06:  LXI     H,128                ;SET UP THE BYTE COUNT
1693  446E  22 40 46          16.0            SHLD    SIZE                 ;
1694                                                                       ;
1695  4471  3E 1B               7.0    TST06X: MVI     A,@033               ;SET -20% W AND -30% R CLOCKS
1696  4473  32 06 43          13.0            STA     SPEED                ;
1697  4476  CD 65 45          18.0            CALL    PESUB                ;CALL SUBROUTINE PESUB
1698  4479  2A 40 46          16.0            LHLD    SIZE                 ;GET THE BYTE COUNT
1699  447C  01 80 00          10.0            LXI     B,128                ;SET UP THE INCREMENT VALUE
1700  447F  09                10.0            DAD     B                    ;INCREMENT THE BYTE COUNT
1701  4480  22 40 46          16.0            SHLD    SIZE                 ;SAVE THE NEW BYTE COUNT
1702  4483  7C                 4.0            MOV     A,H                  ;DO UNTIL THE BYTE COUNT=
1703  4484  FE 03               7.0            CPI     $3
1704  4486  C2 71 44          10.0            JNZ     TST06X
1705  4489  21 58 04          10.0            LXI     H,1112.                      ;SET UP BYTE COUNT
1706  448C  22 40 46          16.0            SHLD    SIZE                         ;
1707  448F  CD 65 45          18.0            CALL    PESUB                        ;CALL SUBROUTINE PESUB
1708  4492                                    ENDTST  TST06
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4492                                      REQ     7                    ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4492  CD 06 28          18.0                               CALL    REQST
 (2)  4495  00                                                   .BYTE           ;DATA PATTERN NUMBER
 (2)  4496  00 00                                                .WORD           ;SYSTEM '#' COUNT
 (2)  4498  00 00                                                .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  449A  00                                                   .BYTE           ;DATA COMPARE FLAG IF -1
 (2)  449B  07                                                   .BYTE   7       ;REQUEST CODE
 (1)  449C  3A 9A 4F          13.0                       LDA     ITERA        ;GET ITERATION COUNT
 (1)  449F  3D                 4.0                       DCR     A            ;DOWNCOUNT
 (1)  44A0  32 9A 4F          13.0                       STA     ITERA        ;SAVE COUNT
 (1)  44A3  F2 6B 44          10.0                       JP      TST06        ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST7 - MTA VELOCITY TESTS

                                ST
                                ; ******************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*MTA - VELOCITY TESTS
                                SD
                                ; ******************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS THE TAPE SPEED, START RAMP, STOP RAMP FORWARD,
                                ;*AND STOP RAMP REVERSE TESTS ON THE SELECTED TU78 WHICH MUST BE IN
                                ;*THE MAINTENANCE MODE.
                                SP
                                ; ******************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    GET THE USER SELECTED TU PORT (MTA)
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    IF MTA IS IN THE MAINTENANCE POSITION
                                ;*    :    THEN - CONTINUE
                                ;*    :    ELSE - ERROR
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE VELTST
                                ;*    IF CARRY SET
                                ;*    :    THEN - ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE RUPTST
                                ;*    IF CARRY SET
                                ;*    :    THEN - ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE FWDTST
                                ;*    IF CARRY SET
                                ;*    :    THEN - ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE REVTST
                                ;*    IF CARRY SET
                                ;*    :    THEN ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*ENDTST
                                SE
                                ; ******************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*
                                S
                                ; ******************************************************************************
```

```
1751  44A6                                  TEST07: TESTX   @7              ;INITIALIZE THE TEST
 (1)  44A6  3E  07              7.0                  MVI     A,@7            ;DEFINE THE TEST NUMBER
 (1)  44A8  CD  03  28          18.0                 CALL    TSET            ;SETUP THE TEST
1752                                    ;%MTA - VELOCITY TESTS
1753                                    ;&VELOCITY ADJUSTMENTS
1754  44AB                              R11L    RIN             ;GET THE MANUAL INTERVENTION BYTE
 (1)  44AB  DB  92              10.0             IN      R11L            ;READ R11L INTO AC
 (1)  44AD  7F                   4.0             MOV     A,A             ;RETRY LINK
1755  44AE  E6  20              7.0      BIT5    ANI                     ;CHECK IF MANUAL INTERVENTION ALLOWED
1756  44B0  CA  62  45          10.0      EXIT    JZ                      ;IF NOT - QUIT
1757  44B3  06  00              7.0      @0      MVI     B,@0            ;ELSE DO VELOCITY TESTS
1758
1759  44B5  3A  3F  46          13.0      UNITMP  LDA                     ;DID THE USER SPECIFY TU
1760  44B8  E6  01              7.0      @01     ANI                     ;PORT 0?
1761  44BA  C2  D0  44          10.0      FOUND1  JNZ                     ;YES - GO USE PORT #0
1762  44BD  04                   4.0              INR     B               ;NO - UPDATE POINTER TO PORT #1
1763  44BE  3A  3F  46          13.0      UNITMP  LDA                     ;DID THE USER SPECIFY TU
1764  44C1  E6  02              7.0      @02     ANI                     ;PORT 1 ?
1765  44C3  C2  D0  44          10.0      FOUND1  JNZ                     ;YES - GO USE PORT #1
1766  44C6  04                   4.0              INR     B               ;NO - UPDATE THE POINTER TO PORT #2
1767  44C7  3A  3F  46          13.0      UNITMP  LDA                     ;DID THE USER SPECIFY TU
1768  44CA  E6  04              7.0      @04     ANI                     ;PORT 2 ?
1769  44CC  C2  D0  44          10.0      FOUND1  JNZ                     ;YES - GO USE PORT #2
1770  44CF  04                   4.0              INR     B               ;NO - ASSUME PORT #3
1771  44D0  CD  13  46          18.0  FOUND1: CALL    CLEAR
1772  44D3  DB  E0              10.0              IN      INTSTA
1773  44D5  E6  80              7.0      BIT7    ANI
1774  44D7  B0                   4.0              ORA     B
1775  44D8  D3  E0              10.0              OUT     MBSEL           ;SELECT DESIRED TU PORT AND MASSBUS
1776
1777  44DA  3E  82              7.0              MVI     A,$82
1778  44DC  D3  40              10.0             OUT     TCMD            ;SELECT MTA REGISTER 2
1779  44DE  DB  40              10.0             IN      TSTS            ;GET THE TU PORT STATUS
1780  44E0  E6  07              7.0              ANI     $07             ;SAVE THE PORT SELECT BITS
1781  44E2  FE  07              7.0              CPI     BIT0!BIT1!BIT2  ;COMPARE TO MAINTENANCE POSITION
1782  44E4  CA  EC  44          10.0              JZ      T2L01           ;YES - CONTINUE TESTING
1783  44E7                                      ERR     EXIT,T2L01
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E7  CD  09  28          18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44EA  02                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44EB  00                                          .BYTE
 (1)  44EC  CD  15  28          18.0      T2L01:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44EF  DA  62  45          10.0              JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1784                            ;>TU78 NOT IN MAINTENANCE POSITION - VELOCITY TESTS ABORTED
1785                            ;>FATAL ERROR - TEST ABORTED
1786
```

```
1788  44F2  CD  5B  00      18.0  VELCOM: CALL    VELTST
1789  44F5  F5                    12.0          PUSH    PSW             ;SAVE THE FLAGS
1790                                        ;     ANA     A               ;MANIPULATE THE FLAGS
1791                                        ;     JP      1$              ;SKIP AROUND IF POSITIVE
1792                                        ;     CMA
1793                                        ;     INR     A               ;COMPLEMENT THE VALUE
1794  44F6                                  1$:   ROUT    ADATA           ;SAVE IT FOR ERROR REPORT
 (1)  44F6  D3  94            10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44F8  7F                 4.0                       MOV     A,A             ;RETRY LINK
1795  44F9  F1                10.0          POP     PSW             ;GET THE FLAGS
1796  44FA  D2  02  45         10.0          JNC     T2L02           ;
1797  44FD                                  ERRA    VELCOM,T2L02
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  44FD  CD  0F  28         18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0003                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4500  03                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4501  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4502  CD  15  28         18.0                 T2L02:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4505  DA  F2  44         10.0                       JC      VELCOM          ;LOOP ADDRESS IF LOOP SPECIFIED
1798                                  ;>TU78 - TAPE SPEED FAILURE
1799
1800  4508  A7                   4.0          ANA     A
1801  4509  CD  5E  00         18.0          CALL    RUPTST
1802  450C  F5                    12.0          PUSH    PSW             ;SAVE THE FLAGS
1803                                        ;     ANA     A               ;MANIPULATE THE FLAGS
1804                                        ;     JP      1$              ;SKIP AROUND IF POSITIVE
1805                                        ;     CMA
1806                                        ;     INR     A               ;COMPLEMENT THE VALUE
1807  450D                                  1$:   ROUT    ADATA           ;SAVE IT FOR ERRORR REPORT
 (1)  450D  D3  94            10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  450F  7F                 4.0                       MOV     A,A             ;RETRY LINK
1808  4510  F1                10.0          POP     PSW             ;GET THE FLAGS
1809  4511  D2  19  45         10.0          JNC     T2L03
1810  4514                                  ERRA    T2L02,T2L03
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  4514  CD  0F  28         18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0004                                        MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4517  04                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4518  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4519  CD  15  28         18.0                 T2L03:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  451C  DA  02  45         10.0                       JC      T2L02           ;LOOP ADDRESS IF LOOP SPECIFIED
1811                                  ;>TU78 - START RAMP FAILURE
1812
1813  451F  A7                   4.0          ANA     A
1814  4520  CD  61  00         18.0          CALL    FWDTST
1815  4523  F5                    12.0          PUSH    PSW             ;SAVE THE FLAGS
1816                                        ;     ANA     A               ;MANIPULATE THE FLAGS
1817                                        ;     JP      1$              ;SKIP AROUND IF POSITIVE
1818                                        ;     CMA
1819                                        ;     INR     A               ;COMPLEMENT THE VALUE
1820  4524                                  1$:   ROUT    ADATA           ;SAVE IT FOR ERROR REPORT
 (1)  4524  D3  94            10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4526  7F                 4.0                       MOV     A,A             ;RETRY LINK
1821  4527  F1                10.0          POP     PSW             ;GET THE FLAGS
```

```
1822  4528  D2  30  45      10.0          JNC     T2L04
1823  452B                                ERRA    T2L03,T2L04
                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)                                                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)  452B  CD  0F  28      18.0                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)        0005                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  452E  05                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  452F  00                           T2L04:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4530  CD  15  28      18.0                    JC      T2L03           ;LOOP ADDRESS IF LOOP SPECIFIED
 (1)  4533  DA  19  45      10.0
1824                                    ;>TU78 - FORWARD STOP RAMP FAILURE
1825
1826  4536  A7                 4.0          ANA     A
1827  4537  CD  64  00       18.0          CALL    REVTST
1828  453A  F5                 12.0          PUSH    PSW             ;SAVE THE FLAGS
1829                                    ;     ANA     A               ;MANIPULATE THE FLAGS
1830                                    ;     JP      1$              ;SKIP AROUND IF POSITIVE
1831                                    ;     CMA
1832  453B  3C                  4.0          INR     A               ;COMPLEMENT THE VALUE
1833  453C                        1$:   ROUT    ADATA           ;SAVE IT FOR ERROR REPORT
 (1)  453C  D3  94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  453E  7F                    4.0                  MOV     A,A             ;RETRY LINK
1834  453F  F1                 10.0          POP     PSW             ;GET THE FLAGS
1835  4540  D2  48  45       10.0          JNC     T2L05
1836  4543                                ERRA    T2L04,T2L05
                                          ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)                                                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)  4543  CD  0F  28      18.0                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)        0006                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4546  06                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4547  00                           T2L05:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4548  CD  15  28      18.0                    JC      T2L04           ;LOOP ADDRESS IF LOOP SPECIFIED
 (1)  454B  DA  30  45      10.0
1837                                    ;>TU78 - REVERSE STOP RAMP FAILURE
1838
1839  454E                                      ENDTST  TST07
                                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (1)                                              REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  454E                                                        CALL    REQST
 (2)  454E  CD  06  28      18.0                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  4551  00                                                    .WORD                   ;SYSTEM '#' COUNT
 (2)  4552  00  00                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4554  00  00                                                .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  4556  00                                                    .BYTE   7               ;REQUEST CODE
 (2)  4557  07
 (1)  4558  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  455B  3D                4.0          DCR     A               ;DOWNCOUNT
 (1)  455C  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  455F  F2  B3  44      10.0          JP      TST07           ;DO TEST UNTIL TILL = 0
1840  4562  C3  18  28      10.0  EXIT:   JMP     TSTEND
```

```
1842                                        .SBTTL  SUBROUTINE PESUB
1843
1844  4565  AF                      4.0   PESUB:  XRA     A               ;CLEAR THE ACCUMULATOR
1845  4566  32  5E  46              13.0           STA     PGCTL           ;SET THE PE FLAG
1846  4569  C3  71  45              10.0           JMP     PEGCOM
1847
1848  456C  3E  FF                  7.0   GCRSUB: MVI     A,@377          ;SET THE GCR FLAG
1849  456E  32  3E  46              13.0           STA     PGCTL
1850
1851  4571  06  00                  7.0   PEGCOM: MVI     B,@0
1852  4573  3A  3F  46              13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1853  4576  E6  01                  7.0           ANI     @01             ;PORT 0?
1854  4578  C2  8E  45              10.0           JNZ     FOUND           ;YES-GO USE PORT #0
1855  457B  04                      4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
1856  457C  3A  3F  46              13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1857  457F  E6  02                  7.0           ANI     @02             ;PORT 1?
1858  4581  C2  8E  45              10.0           JNZ     FOUND           ;YES-GO USE PORT #1
1859  4584  04                      4.0           INR     B               ;NO-UPDATE POINTER TO PORT #2
1860  4585  3A  3F  46              13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1861  4588  E6  04                  7.0           ANI     @04             ;PORT 2?
1862  458A  C2  8E  45              10.0           JNZ     FOUND           ;YES-GO USE PORT #2
1863  458D  04                      4.0           INR     B               ;NO-ASSUME PORT #3
1864  458E  CD  13  46              18.0   FOUND:  CALL    CLEAR
1865  4591  DB  E0                  10.0           IN      INTSTA
1866  4593  E6  80                  7.0           ANI     BIT7
1867  4595  B0                      4.0           ORA     B
1868  4596  D3  E0                  10.0           OUT     MBSEL           ;SELECT THE DESIRED TU-PORT AND MASSBUS
1869
1870  4598  AF                      4.0   T1L01:  XRA     A               ;CLEAR THE DATA BUS CONTROL
1871  4599  32  28  4F              13.0           STA     DBUS            ;MASK TO KEEP THE MASSBUS CLEAN
1872                                                                       ;
1873  459C  2A  40  46              16.0   LHLD    SIZE                    ;GET THE RECORD SIZE
1874  459F  22  23  4F              16.0   SHLD    BYTEL                   ;TO 128 BYTES
1875  45A2  7D                      4.0   MOV     A,L                     ;GET THE BYTE COUNT LOW
1876  45A3                                ROUT    RO5L                    ;SAVE IN BYTE COUNT LOW
 (1)  45A3  D3  8A                  10.0           OUT     RO5L                    ;WRITE AC INTO RO5L
 (1)  45A5  7F                      4.0           MOV     A,A                     ;RETRY LINK
1877  45A6  7C                      4.0   MOV     A,H                     ;GET THE BYTE COUNT HIGH
1878  45A7                                ROUT    RO5H                    ;SAVE IN BYTE COUNT HIGH
 (1)  45A7  D3  88                  10.0           OUT     RO5H                    ;WRITE AC INTO RO5H
 (1)  45A9  7F                      4.0           MOV     A,A                     ;RETRY LINK
1879                                                                       ;
1880  45AA  3A  42  46              13.0   LDA     FORMT                   ;GET THE FORMAT
1881  45AD  32  25  4F              13.0   STA     FORMAT                  ;NO SKIP COUNT
1882  45B0                                ROUT    RO2H                    ;SAVE IN THE CAS
 (1)  45B0  D3  85                  10.0           OUT     RO2H                    ;WRITE AC INTO RO2H
 (1)  45B2  7F                      4.0           MOV     A,A                     ;RETRY LINK
1883                                                                       ;
1884  45B3  3E  88                  7.0   MVI     A,DDRCO                 ;CLEAR DDR IN, SO DDR
1885  45B5  D3  DB                  10.0   OUT     DDRCTL                  ;WRITES TO TAPE
1886                                                                       ;
1887  45B7  2A  43  46              16.0   LHLD    KWDAT                   ;GET THE WRITE DATA
1888  45BA  29                      10.0   DAD     H                       ;SHIFT IT LEFT 1 BIT POSITION
1889  45BB  17                      4.0   RAL                             ;SAVE THE CARRY
```

```
1890  45BC  29          10.0        DAD   H          ;SHIFT IT LEFT ANOTHER BIT POSITION
1891  45BD  17           4.0        RAL              ;SAVE THAT CARRY TOO.
1892  45BE  E6 03        7.0        ANI   $3         ;SAVE ONLY THE CARRY BITS
1893  45C0  47           4.0        MOV   B,A        ;SAVE IN REGISTER B
1894  45C1  7C           4.0        MOV   A,H        ;
1895  45C2  D3 D9       10.0        OUT   DDRB       ;LOAD <13:6> IN DDR B
1896  45C4  3A 45 46    13.0        LDA   KWDAT+2    ;GET BITS <17:16>
1897  45C7  85           4.0        ADD   L          ;MERGE WITH <5:0>
1898  45C8  D3 D8       10.0        OUT   DDRA       ;LOAD INTO DDR A
1899  45CA  AC           4.0        XRA   H          ;COMPUTE PARITY ON ALL BITS
1900  45CB  A8           4.0        XRA   B          ;
1901  45CC  78           4.0        MOV   A,B        ;GET BITS <15:14> AGAIN
1902  45CD  E2 D2 45     10.0        JPO   1$         ;JUMP IF ODD PARITY
1903  45D0  F6 04         7.0        ORI   BIT2       ;SET THE PARITY BIT TO MAKE IT ODD
1904  45D2  D3 DA        10.0  1$:   OUT   DDRC       ;LOAD <P:14> IN DDRC
1905                                                  ;
1906  45D4  21 03 43     10.0        LXI   H,SPEED-3  ;POINT TO THE CLOCK
1907  45D7  22 26 4F     16.0        SHLD  CLOCK      ;SPEED TABLE
1908  45DA  3E 00         7.0        MVI   A,0
1909  45DC  D3 D2        10.0        OUT   TRKENA
1910  45DE  D3 D2        10.0        OUT   TRKENA
1911  45E0  3E FF         7.0        MVI   A,@377
1912  45E2  D3 D2        10.0        OUT   TRKENA
1913  45E4  D3 D2        10.0        OUT   TRKENA
1914  45E6  3A 3E 46     13.0        LDA   PGCTL      ;GET THE PE/GCR FLAG
1915  45E9  A7           4.0        ANA   A          ;SET CONDITION BITS
1916  45EA  C2 F3 45     10.0        JNZ   2$         ;GO DO GCR
1917  45ED  CD 55 00     18.0        CALL  LKLWMP     ;ELSE-DO PE
1918  45F0  C3 F6 45     10.0        JMP   3$         ;
1919  45F3  CD 58 00     18.0  2$:   CALL  LKLWMG     ;CALL LOOP WRITE/READ - MTA - GCR
1920                                                  ;
1921  45F6  D2 0C 46     10.0  3$:   JNC   T1CN1                ;CONTINUE IF NO ERROR
1922  45F9  78           4.0        MOV   A,B                  ;GET THE FAILURE CODE
1923  45FA  E6 FC         7.0        ANI   $FC                  ;MASK OUT THE MEANINGFUL BITS
1924  45FC  0F           4.0        RRC                        ;POSITION FOR OUTPUT
1925  45FD  0F           4.0        RRC                        ;
1926  45FE                          ROUT  ADATA                ;STORE IN THE CAS
 (1)  45FE  D3 94        10.0              OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  4600  7F           4.0              MOV   A,A            ;RETRY LINK
1927  4601  79           4.0        MOV   A,C                  ;GET THE INTERRUPT CODE
1928  4602  E6 3F         7.0        ANI   $3F                  ;MASK OUT THE MEANINGFUL BITS
1929  4604                          ROUT  EDATA                ;STORE IN THE CAS
 (1)  4604  D3 95        10.0              OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  4606  7F           4.0              MOV   A,A            ;RETRY LINK
```

```
1931  4607                                          ERRB    TEST07,T1CN1,3
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA 'ADATA' AND EXPECTED DATA 'EDATA' VALID
 (1)  4607  CD   12   28        18.0                        CALL    ERL^B           ;PROCESS ERROR - DO 2.3
 (1)        0007                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460A  07                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  460B  03                                            .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  460C  CD   15   28        18.0                T1CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  460F  DA   A6   44        10.0                        JC      TEST07          ;LOOP ADDRESS IF LOOP SPECIFIED
1932                                            ;>LOOP WRITE/READ ERROR
1933                                            ;<ACTUAL DATA IS THE FAILURE CODE
1934                                            ;<EXPECTED DATA IS THE INTERRUPT CODE
1935  4612  C9                10.0                        RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL MTAS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL MTAS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL MTA PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************

 4613 F5            12.0  CLEAR:  PUSH  PSW             ;SAVE THE SELECTED PORT #
 4614 C5            12.0          PUSH  B
 4615 06 00          7.0          MVI   B,0             ;START TO CLEAR AT PORT #0
 4617 DB E0         10.0  CLRLP:  IN    INTSTA          ;GET MB SELECT INFO
 4619 E6 80          7.0          ANI   BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
 461B B0             4.0          ORA   B               ;ADD IN THE SELECTED PORT #
 461C D3 E0         10.0          OUT   MBSEL           ;RESET TO THIS PORT
 461E 3E 80          7.0          MVI   A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
 4620 D3 40         10.0          OUT   TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
 4622 AF             4.0          XRA   A               ;CLEAR TU COMMAND A
 4623 D3 40         10.0          OUT   TCMD            ;
 4625 3E 81          7.0          MVI   A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
 4627 D3 40         10.0          OUT   TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
 4629 3E 00          7.0          MVI   A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
 462B D3 40         10.0          OUT   TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
 462D AF             4.0          XRA   A
 462E D3 44         10.0          OUT   TAMT            ;CLEAR AMTIE WORD
 4630 D3 48         10.0          OUT   PDIAG           ;CLEAR DIAG CONTROL WORD
 4632 D3 4C         10.0          OUT   PENAB           ;CLEAR PORT ENABLE WORD
 4634 04             4.0          INR   B               ;POINT TO THE NEXT PORT TO CLEAR
 4635 78             4.0          MOV   A,B
 4636 FE 04          7.0          CPI   4               ;DONE?
 4638 C2 17 46      10.0          JNZ   CLRLP           ;NO - CLEAR THIS PORT ALSO
 4638 C1            10.0          POP   B               ;RESET B & C
 463C F1            10.0          POP   PSW             ;ALL DONE
 463D C9            10.0          RET                   ;EXIT
```

```
                                        .SBTTL  PROGRAM VARIABLES

1982
1983
1984  463E  00                          PGCTL:  .BYTE   0                ;PE/GCR CONTROL FLAG
1985  463F  00                          UNITMP: .BYTE   0                ;UNIT MAP
1986  4640  00  00                      SIZE:   .WORD   0                ;RECORD SIZE
1987  4642  00                          FORMT:  .BYTE   0                ;DATA FORMAT
1988  4643  00                          KWDAT:  .BYTE   0                ;18 BITS OF INPUT DATA
1989  4644  00                                  .BYTE   0
1990  4645  00                                  .BYTE   0
1991        0000                                .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 0089          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLEAR   4613           CLRCTL= 00F0           CLOCK   4F26          CLRLP   4617
CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAVE   4F9D           CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0C85
CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082          C.     = 0001
C.AVAI= 0008           C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002          D      =%0002          DATACT= 00D0
DBUS    4F28           DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300           DIAGRM= 4F90           DIARD  = 0008          DONE1  = 0045
DONINT= 0010           DSAVE   4F9E           DSE    = 0006          DUMMY   4325   G
D.ATH0= 0001           D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020
D.NOTW= 0040           D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E           EDATA  = 0095          EOTCLR= 0003
ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812
ERLPE  = 280C          ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F
EXIT    4562           E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25
FORMT   4642           FOUND   458E           FOUND1  44D0          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002           GCRSUB  456C          GOODTM= 0092
H      =%0004          HLSAVE  4FAC           IE     = 0008          INTSTA= 00E0
ITERA   4F9A           I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F          KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D          KEY11  = 006E          KEY12  = 006F
KEY13  = 005C          KEY14  = 005D          KEY15  = 005E          KEY16  = 005F
KEY17  = 003C          KEY18  = 003D          KEY19  = 003E          KEY2   = 0079
KEY20  = 003F          KEY3   = 007A          KEY4   = 007B          KEY5   = 0074
KEY6   = 0075          KEY7   = 0076          KEY8   = 0077          KEY9   = 006C
KINTA  = 006F          KLDAD  = 003D          KN0    = 003C          KN1    = 005C
KN2    = 005D          KN3    = 005E          KN4    = 006C          KN5    = 006D
KN6    = 006E          KN7    = 0074          KN8    = 0075          KN9    = 0076
```

```
KU2    = 0079          KU3    = 007A          KU8    = 0077          KWDAT   4643
L      =%0005          LBLANK= 000F           LCE    = 0008          LCH    = 000C
LCL    = 000D          LCP    = 000E          LC0    = 0000          LC1    = 0001
LC2    = 0002          LC3    = 0003          LC4    = 0004          LC5    = 0005
LC6    = 0006          LC7    = 0007          LC8    = 0008          LC9    = 0009
LDLEDA= 00CA           LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046          LPFLG   4F94
LPNUM   4F92           M      =%0006          MBSEL  = 00E0          MB.A   = 0008
MB.B   = 0004          MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000
MSE    = 0008          MSGN   = 0007          MTACLR= 0000           MT.ARA= 0020
MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008
MT.LWR= 0004           MT.MGT= 0002           MT.NWT= 0080           MT.PEC= 0040
MT.PSB= 0004           MT.PS0= 0001           MT.PS1= 0002           MT.REV= 0020
MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020
M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008
M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020
M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080           M.RDEN= 0002
M.RDPE= C008           M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040
M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080
M5.5   = 0001          M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088
OKAY   = 00FF          OPRRAM= 4300           OPSTRT= 0058           OPVER  = 0040
PADCNT= 00D5           PADCRC= 0080           PDIAG  = 0048          PEGCOM  4571
PEID   = 008A          PENAB  = 004C          PESET  = 0001          PESUB   4565
PGCTL   463E           PL     = 00B1          PRDD   = 004C          PRENF  = 009C
PS     = 0082          PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001
P.BCTC= 0040           P.CMDP= 0020           P.INTE= 0080           P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004           P.RPST= 0002
P.RPOE= 0020           P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010
P.SING= 0080           P.STAT= 0002           P.STPE= 0080           P.TACH= 0008
P.TUPR= 0010           P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010           P.WPOE= 0008           P.WP1E= 0004           P.WP2E= 0008
P.WP3E= 0004           P.5VOK= 0002           QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047           RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          REND   = 0014
REQST  = 2806          RESCHR= 00D1           REVTST= 0064           REWIND= 0004
RFIFOL= 0008           RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001
RILL   = 0012          RINST  = 000C          RMCTST= 0008           RMK2   = 0013
RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001
RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000
RPF1   = 009D          RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016
RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A
RTIER  = 0030          RTM    = 0005          RUNKI  = 0009          RUPTST= 005E
RWDUNL= 0005           R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040
R.DON  = 0002          R.DRDY= 0010           R.END  = 0010          R.ILL  = 0004
R.JVOK  0004           R.MK2  = 0008          R.PLOD= 0008           R.PLOO= 0010
R.PLO1= 0020           R.POST= 0020           R.STNM= 0002           R.STOP= 0004
R.STPC= 0001           R.TBJN= 0080           R.TSTD= 0040           R.VOK  = 0080
R00H   = 0081          R00L   = 0080          R01H   = 0083          R01L   = 0082
R02H   = 0085          R02L   = 0084          R03H   = 0087          R03L   = 0086
R04H   = 0089          R04L   = 0088          R05H   = 008B          R05L   = 008A
```

```
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SIZE    4640          SOD   = 0080          SOE   = 0040          SP    =%0008
SPEED   4306          SSCLK = 0040          SSTEP = 0005          STACK = 4FFF
STATRM= 4F20          STPCT   4F20          STRSP = 5000          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST02  4366          TEST03  43A6          TEST04  43E6          TEST05  4426
TEST06  4466          TEST07  44A6          TEST1   4307          TMF   = 0099
TMRDY = 0040          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TST01X  432B          TST01Y  4331          TST02   4368
TST02X  4371          TST03   43AB          TST03X  43B1          TST04   43EB
TST04X  43F1          TST05   442B          TST05X  4431          TST06   4468
TST06X  4471          TST07   44B3          TUSEL0= 00D1          TUSEL1= 00D2
TU78  = 0010          T.ATH0= 0001          T.ATH1= 0002          T.FSOT = 0004
T.EOT = 0002          T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020
T.PES = 0008          T.PSBJ= 0020          T.PS0J= 0008          T.PS1J= 0010
T.RDY = 0080          T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002
T1CN1   460C  G       T1L01   4598          T2L01   44EC  G       T2L02   4502  G
T2L03   4519  G       T2L04   4530  G       T2L05   4548  G       UIBG  = 00A1
UNITMP  463F          VALFC   4F98          VALTB   4F95          VELCOM  44F2
VELTST= 005B          WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA
WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004
W.CRC = 0008          W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010
W.ENAB= 0080          W.ERR = 0020          W.FMT = 0070          W.GCR = 0010
W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002          W.REV = 0004
W.ROME= 0010          W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008
W.XFER= 0020          X     =%000A          X.DONN= 0080          X.ENAB= 0040
X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001          Y     =%0008
.     = 4646
```

ERRORS DETECTED: 0

*MTA4.A78/PTP,MTA4=NLIST,PARAM,MACRO,LIST,MTA4
RUN-TIME: 4 4 0 SECONDS
CORE USED: 10K