DUV-11

OFFLINE RECEIVER TIMING TEST
MD-11-DZDUS-B

EP-DZDUS-B-DL-B

FICHE 1 OF 1

DEC 1977
digital
MADE IN USA

```
.REM    *
```

I D E N T I F I C A T I O N

PRODUCT CODE: MAINDEC-11-DZDUS-B-D

PRODUCT NAME: DUV11 OFFLINE RECEIVER TIMING TESTS

RELEASE DATE: NOV. 1977

MAINTAINER : DIAGNOSTICS

```
*
.REM    *
```



```
*
```

```
                                                        .REM   *
```

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DUV11'S. THIS MEANS THAT
16 DEVICES CAN BE SEQUENTIALLY EXERCISED. THE DIAGNOSTIC
MAKES ONE PASS BEFORE PROCEEDING TO THE NEXT DEVICE,
AND CONTINUES EXERCISING ALL DEVICES IN THIS FASHION UNTIL
HALTED.

```
                                                        *
                                                              .REM    *
```

1. THE DUV11 OFFLINE RECEIVER TIMING TESTS VERIFY THAT THE
RECEIVER LOGIC AND ASSOCIATED ERROR FLAGS ASSERT
AT THE PROPPER TIME

```
                                                        *
                                                              .REM    *
```

2. REQUIREMENTS

PDP-11/03 COMPUTER (LSI)

DUV11 SYNCRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE
THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

3. LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES
IS TO BE USED.

	STARTING ADDRESS FOR ABSOLUTE LOADER
4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

NOTE: ALL SWITCHES RESIDE INTERNAL TO THE CPU AT ADDRESS
176. THESE MAY BE SET VIA THE CONSOLE TTY BY DIRECTLY
MODIFYING LOC. 176.

NOTE: RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER
CALLED "$USWR". IN ORDER TO BE FLEXIBLE ON THE AVAILIBILITY OF THE
H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11.
BIT 0 IN $USWR REFLECTS THIS STATUS. A 0 = CONNECTOR
PRESENT, A 1 = CONNECTOR NOT AVAILIBLE.

THE USER CHANGES THE CONTENTS OF THIS LOCATION
WHEN BUILDING THE E TABLE, BY ANSWERING THE
PROMPT "SWITCH 2".

4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)
ALL CONSOLE SWITCHES DOWN

4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)
SW01=1

4.1.4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)

SW14=1
NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW14=1 IS USED
NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RETARTING ADDRESS FOR ALL TESTS IS 000200
THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200
THE STARTING ADDRESS TO LOCK ON TEST IS 000200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4.3.1.2 SET SWITCH REGISTER (LOC. 176) TO ZERO.

4.3.1.3 TYPE 200G.

4.3.1.4 PROGRAM WILL START.

*
.REM *

4.3.1.5 THE PROGRAM WILL TYPE "DUV11 DZDUS-B TAPE C" (ONCE ONLY)

*
.REM *

4.3.1.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT
TO START TESTING ,AND THEN TESTING WILL BEGIN

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 THE PROGRAM WILL TYPE "R" AND WILL COMMENCE TESTING

4.3.3 PROGRAM RESTART WITH SW00=1

4.3.3.1 SET SWITCH REGISTER (LOC. 176) TO A 000001.

4.3.3.2 TYPE 200G.

4.3.3.3 PROGRAM WILL START.

4.3.3.4 THE PROGRAM WILL TYPE " 1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.5 TYPE IN THE ADDRESS OF THE FIRST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.4

4.3.3.6 THE PROGRAM WILL TYPE "VECTOR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.7 TYPE IN THE BASE RECEIVER INTERRUPT VECTOR ADDRESS FOR THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.6

4.3.3.8 THE PROGRAM WILL TYPE "ARE YOU RUNNING MULTIPLE DEVICES ?" (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.9 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS GIVEN, THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.8

IF A "NO" ANSWER IS GIVEN: JUMP TO SECTION 4.3.3.12
IF A "YES" ANSWER IS GIVEN: THE NEXT QUESTION IS ASKED

4.3.3.10 THE PROGRAM WILL TYPE "LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.10
NOTE: ALL ADDRESSES SHALL BE CONTIGUOUS

4.3.3.11.1 IF AN "OUT OF RANGE" ADDRESS IS TYPED IE. MORE THAN 16 (10) DEVICES AWAY (UPWARDS)..... THE PROGRAM WILL TYPE "OUT OF RANGE: RETYPE LAST DEVICE RXCSR ADDRESS-"

AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.11.1

IF A DEVICE ADDRESS LOWER THAN 1ST DEVICE ADDRESS IS TYPED.....
....SCHOOLS OUT.....THERE IS NO PROTECTION FOR THIS.
THE PROGRAM WILL DEFAULT TO TWO DEVICES ACTIVE (UPWARDS FROM 1ST DEVICE ADDRESS). THE SAME APPLIES TO IDENTICAL ADDRESSES TYPED FOR FIRST AND LAST DEVICE.
OBSERVE LOCATION @ ACTREG: SEE SECTION 7.2

4.3.3.12 THE PROGRAM WILL TYPE "# OF SYNC CHARS SELECTED (1 OR 2)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD. REFER TO MANUAL FOR PROPER SWITCH SETTINGS OF SWITCH E55-4.

4.3.3.13 TYPE IN THE APPROPRIATE ANSWER "1" OR "2" FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.12

4.3.3.14 THE PROGRAM WILL TYPE " IS SEC XMIT SWITCH E55-2 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.15 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE THAT ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.14

4.3.3.16 THE PROGRAM WILL TYPE "IS SEC REC SWITCH E55-3 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.17 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.16

4.3.3.18 THE PROGRAM WILL TYPE "IS OPT CLR ENABLE SWITCH E55-1 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.19 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED

BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.18

4.3.3.20 THE PROGRAM WILL TYPE "ARE YOU RUNNING IN MAINT.
MODE EXTERNAL ? AND .....DO YOU HAVE THE EXTERNAL MODEM
BYPASS JUMPER CONNECTOR ON ? (Y OR N)-" AND WAIT FOR AN
INPUT FROM THE TELETYPE KEYBOARD

4.3.3.21 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY
A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.20

4.3.3.22 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT
HAS STARTED AND WILL COMMENCE TESTING AT TEST 1

4.3.4 PROGRAM RESTART WITH SW01=1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
,,,IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED

IF MULTIPLE DEVICES WERE PREVIOUSLY SELECTED,LOAD 000200,
AND SELECT SW00=1 AND ANSWER "NO" TO THE MULTIPLE DEVICE QUESTION
SEE 4.3.3

4.3.4.1 SET SW01=1 IN SWITCH REG (LOC. 176)

4.3.4.2 TYPE 200G.

4.3.4.3 PROGRAM WILL START.

4.3.4.4 THE PROGRAM WILL TYPE "TEST PC-" AND WAIT FOR AN INPUT FROM
THE TELETYPE KEYBOARD

4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO
BE STARTED FOLLOWED BY A <CARRIAGE RETURN>

4.3.4.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT THE SELECTED TEST

NOTE: CARE MUST BE TAKEN WHEN THIS FEATURE IS USED
,SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS
THAT IS IN THE MIDDLE OF A TEST

4.3.5 PROGRAM RESTART WITH SW14 =1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 SET SW14=1 IN SWITCH REG. (LOC. 176)

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

4.3.5.4 THE PROGRAM WILL TYPE "LOCK ON SELECTED TEST ? (Y OR N)-"
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.5.5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A
<CARRIAGE RETURN>

IF A NO ANSWER IS GIVEN: THIS LOCK ON TEST WILL BE IGNORED
AND THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1

4.3.5.6 IF A YES ANSWER WAS GIVEN: THE PROGRAM WILL ACT AS FOLLOWS...
THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1 AND WILL REMAIN IN TEST 1 UNTIL HALTED
OR IF ANY KEY IS STRUCK ON THE TELETYPE ,THE PROGRAM
WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON
THE TELETYPE AND SO FORTH THRU THE PROGRAM. IF SW01 =1 IT
WILL PERFORM AS IN SECTION 4.3.4 ALLOWING ONE TO FREEZE
ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS (INTERNAL TO THE CPU, ACCESSED VIA LOC. 176).

SW15 =1	HALT ON ERROR
SW14 =1	LOOP ON CURRENT TEST
SW13 =1	INHIBIT ERROR TYPEOUT
SW11 =1	INHIBIT ITERATIONS
SW10 =1	ESCAPE TO NEXT TEST ON ERROR
SW09 =1	LOOP ON ERROR
SW01 =1	RESTART PROGRAM AT SELECTED TEST
SW00 =1	RESELECT VECTOR AND CONTROL REGISTER ADRESSES &PARAMETERS AFTER A PROGRAM RESTART

TO INHIBIT "END OF PASS" TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS (UNDER LSI ALL HALT ERRORS RETURN CONTROL TO O.D.T.)
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

6.1.1 PC+2 = ERROR PC
WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2

REFER TO THE ABOVE "HLT" IN DIAGNOSTIC FOR ERROR DESCRIPTION

CHECK ADDRESS @ RXCSR: TO LOCATE THE DEVICE PRESENTLY UNDER
TEST WHEN RUNNING MULTIPLE DEVICES

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING RECEIVER (RXDBUF) REGISTER

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING TRANSMITTER (TXCSR) REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15 =0
IF THE PROGRAM IS RUN WITH SW15 =0 ,NO OPERATOR ACTION IS
REQUIRED TO CONTINUE TESTING

6.2.2 SW15 =1
IF THE PROGRAM IS RUN WITH SW15 =1 ,TO CONTINUE TESTING
AFTER THE PROGRAM HAS HALTED ,PRESS THE PROCESSOR
CONSOLE "CONTINUE SWITCH"

NOTE: THE PC + 2 OF THE "HLT" WILL BE DISPLAYED IN THE DATA LIGHTS

6.2.3 ILLEGAL INTERRUPTS
IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED
DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN
THE TRAPCATCHER. THE ADDRESS AT WHICH THE PROGRAM
HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT
OCCURED. THE PROGRAM MUST BE RESTARTED AT 000200 TO
RECOVER FROM THIS ERROR.

6.2.4 ADDITIONAL TROUBLESHOOTING AIDS ERRCNT: & PASCNT:
CHECK THESE TWO TAG LOCATIONS FOR TOTAL # OF ERRORS AND PASSES PESPECTIVELY.
LOADING 000200 AND RESTARTING WILL CLEAR THESE LOCATIONS.

6.3 END OF PASS ROUTINE
THIS TYPEOUT IS MENTIONED HERE FOR CONVENIENCE
IT IS IN THE FORM:

```
                END OF PASS TAPE Y
                16XXXX = DEVICE

                WHERE Y IS THE TAPE LOADED

                WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

                TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

        7.      RESTRICTIONS

        7.1     MULTIPLE DEVICES
                UP TO 16(10) DEVICES MAY BE TESTED. HOWEVER, THEY
                MUST HAVE CONTIGUOUS ADDRESSES AND VECTORS

                NOTE: IF ALL DEVICES UNDER TEST HAVE THE SAME INTERRUPT VECTOR
                YOU CAN CHANGE  "ZERO:    ADD   #10,BASEIV      ;NEXT BLOCK
                (VECTORS)" TO "ZERO:   ADD    #0,BASEIV";
                THEREBY THE VECTOR ADDRESSES WILL NOT BE
                UPDATED AFTER EACH PASS.

        7.2     DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

                WHEN RUNNING MULTIPLE DEVICES    AN ACTIVE BIT IS SET
                FOR EACH DEVICE RUNNING UNDER TEST  IE. BIT 0 FOR
                DEVICE 0   .BIT 15 FOR DEVICE 15
                TO DISQUALIFY DEVICES:

        7.2.1   IF DEVICE 0 IS TO BE DISQUALIFIED ,SIMPLY RESTART
                PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.

        7.2.2   IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF
                ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG:
                OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)
                AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

                7.2.2.1 TO RESTART...TYPE 200G...
                        THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

                7.2.2.2 .......OR .....SET SW00=1 IN SWITCH REG (LOC. 176) AND TYPE 200G....
                        ANSWER THE QUESTION : 1ST DEVICE : ETC.....
                         ....THE PROGRAM WILL CONTINUE WITH DEVICE 0

                7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE   THE PROGRAM
                        WILL TYPEOUT   AN ERROR MESSAGE.......TYPE 200G.

        7.3     CABLE DELAYS
                NOTE: EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH H315 CONNECTOR ON)

        7.3.1   TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,
                LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.
                        PRESENTLY "HOLD:" =20  IS SUFFICIENT TIME ON AN 11/03 MACHINE.

                BASICALLY DON'T TRY TO EXCEED 10K TO 12K RATE USING THE EIA DRIVERS

        7.4     TO USE THE "XOR" TESTER ,THE BRANCH AROUND THE "XOR"
```

CODE MUST BE PATCHED TO A "NOP". (SEE LISTINGS FOR DETAILS)

8. DEFAULT PARAMETERS:
1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160010

VECTOR ADDRESS- DURIV: 770

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0

LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

OF SYNC CHARS SELECTED - 2 SYNCNO: 377

IS SEC XMIT SWITCH E55-2 ON?- YES SEXMIT: 377

IS SEC REC SWITCH E55-3 ON?- YES SEREC: 377

IS OPT CLR ENABLE SWITCH E55-1 ON?- YES OPTCLR: 377

DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
CONNECTOR ON (H315)- YES JMRBY: 377

9. PROGRAM DESCRIPTION

*
.REM *

9.1 THIS PROGRAM PERFOFMS THE OFFLINE RECEIVER TIMING TESTING
OF THE DEVICE
SEE LISTING FOR DETAILS

*
.REM *

10. FLOW CHARTS: RECEIVER FLOW, TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW

11. LISTINGS

*

```
524
525
526
527
528
529
530
531
532
533
534
535
536             000001                  $TN=1
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
```

```
                                .ENABLE ABS

                                ;DUV11 DZDUS-B TAPE C
                                ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;TYPE 200G
                                        ;PROGRAM WILL TYPE "DUV11 DZDUS-B TAPE C "
                                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE C"
                                        ;AND THEN RESUME TESTING

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
000006                          SP=     %6              ;;STACK POINTER
000007                          PC=     %7              ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
```

```
                020000                  SW13=   20000
                010000                  SW12=   10000
                004000                  SW11=   4000
                002000                  SW10=   2000
                001000                  SW09=   1000
                000400                  SW08=   400
                000200                  SW07=   200
                000100                  SW06=   100
                000040                  SW05=   40
                000020                  SW04=   20
                000010                  SW03=   10
                000004                  SW02=   4
                000002                  SW01=   2
                000001                  SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                100000                  BIT15=  100000
                040000                  BIT14=  40000
                020000                  BIT13=  20000
                010000                  BIT12=  10000
                004000                  BIT11=  4000
                002000                  BIT10=  2000
                001000                  BIT09=  1000
                000400                  BIT08=  400
                000200                  BIT07=  200
                000100                  BIT06=  100
                000040                  BIT05=  40
                000020                  BIT04=  20
                000010                  BIT03=  10
                000004                  BIT02=  4
                000002                  BIT01=  2
                000001                  BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
```

```
   670          000014                  TBITVEC=14              ;;"T" BIT
   671          000014                  TRTVEC= 14              ;;TRACE TRAP
   672          000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
   673          000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
   674          000024                  PWRVEC= 24              ;;POWER FAIL
   675          000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
   676          000034                  TRAPVEC=34              ;;"TRAP" TRAP
   677          000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
   678          000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
   679          000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                        ;STANDARD INTERRUPT VECTORS


                000174                  .=174
000174  000000                         DISPREG: 0
000176  000000                         SWREG: 0
                000200                  .=200
000200  000167  001746                  JMP     .START          ;GO TO START OF PROGRAM


                001100                  .=1100
001100  000000                         .WORD 0
001102  177570                         LIGHTS: 177570


                                        ;PROGRAM CONTROL PARAMETERS

001104  000000                         RETURN: 0
001106  000000                         NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001110  000000                         LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
001112  000000                         PASCNT: 0               ;ADDRESS CONTAINING PASS COUNT
001114  000000                         ERRCNT: 0                       ;ERROR COUNT
001116  000000                         SAVSP:  0               ;STACK POINTER STORAGE

                                        ;PROGRAM VARIABLES

001120  000020                         HOLD:   20              ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
001122  000000                         SHIFT:  0               ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
001124  000000                         COUNT:  0               ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
001126  000000                         SAVPC:  0                       ;PROGRAM COUNTER STORAGE
001130  000000                         HLD0:   0
001132  000000                         HLD1:   0
001134  000000                         HLD2:   0
001136  000000                         HLD3:   0
001140  000000                         HLD4:   0
001142  000000                         HLD5:   0
001144  000000                         HLD6:   0
```

```
                                        ;PROGRAM CONVERSATIONAL PARAMETERS
001146      377         SYNCNO: .BYTE   377     ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
001147      377         SEXMIT: .BYTE   377     ;SEC XMIT JUMPER "IN"
001150      377         SEREC:  .BYTE   377     ;SEC REC JUMPER "IN"
001151      377         OPTCLR: .BYTE   377     ;OPTIONAL JUMPER CLR "IN"
001152      000         MULTD:  .BYTE   0       ;NO MULTIPLE DEVICE FLAG
001153      377         JMRBY:  .BYTE   377     ;EXTERNAL MODEM BYPASS JUMPER "IN"
                        .EVEN

                                        ;PROGRAM MULTIPLE DEVICE PARAMETERS
001154  000000          BASEADD:        0       ;PROG CONTROLLED 1ST DEVICE ADDR
001156  000000          KEEPADD:        0       ;SAVED 1ST DEVICE ADDR
001160  000000          LASTADD:        0       ;LAST DEVICE RXCSR ADDR
001162  000000          BASEIV:         0       ;PROG CONTROLLED IV
001164  000000          KEEPIV:         0       ;SAVED INTR VECTOR
001166  000000          ACTREG:         0       ;ACTIVE REGISTER ,,,MODIFY THIS
                                                ;LOCATION TO DISQUALIFY OR QUALIFY
                                                ;DEVICES (1= RUN,,,0= DON'T RUN)
001170  000000          ROTADD:         0       ;ROTATING POINTER FOR ACTREG..POINTS
                        ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES

                                        ;PROGRAM CONTROL FLAGS

001172      000         INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001173      000         STFLG:  .BYTE   0               ;TEST START FLAG
001174      000         LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
            001176      .EVEN
            001400      .=1400
```

```
                        ;INSTRUCTION DEFINITIONS

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                  PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                  POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                  POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                        ;REGISTER DEFINITIONS
                                ;RXCSR BIT DEFINITIONS
100000                  DSC=BIT15               ;DATA SET CHANGE
040000                  RING=BIT14              ;RING
020000                  CTS=BIT13               ;CLR TO SEND
010000                  CARDET=BIT12            ;CARRIER DETECT
004000                  RECACT=BIT11            ;REC ACTIVE
002000                  SRD=BIT10               ;SEC REC DATA
001000                  DSR=BIT9                ;DATA SET RDY
000400                  STPSYN=BIT8             ;STRIP SYNC
000200                  RXDONE=BIT7             ;REC DONE
000100                  RINTEN=BIT6             ;REC INTR ENABLE
000040                  DSINTE=BIT5             ;DSC INTR ENABLE
000020                  SYNSCH=BIT4             ;SYNC SEARCH
000010                  STD=BIT3                ;SEC XMIT DATA
000004                  RTS=BIT2                ;REQ TO SEND
000002                  DTR=BIT1                ;DATA TERM RDY
000001                  VOID=BIT0
                                ;RXDBUF BIT DEFINITIONS
100000                  RXERR=BIT15             ;REC ERROR
040000                  OVRRUN=BIT14            ;OVERRUN
020000                  FRMERR=BIT13            ;FRAME ERROR
010000                  PARER=BIT12             ;PARITY ERROR
                                ;PARCSR BIT DEFINITIONS
001000                  PAREN=BIT9              ;PARITY ENABLE
000400                  EVPAR=BIT8              ;EVEN PARITY SENSE
                                ;PARCSR WRD DEFINITIONS
030000                  SYNINT=30000            ;SYNC EXTERNAL MODE
020000                  SYNEXT=20000            ;SYNC INTERNAL MODE
000000                  ISYMOD=0                ;ISOC MODE
000000                  FIVE=0                  ;WORD LENGTH 5 BITS
002000                  SIX=2000                ;WORD LENGTH 6 BITS
004000                  SEVEN=4000              ;WORD LENGTH 7 BITS
006000                  EIGHT=6000              ;WORD LENGTH 8 BITS
000000                  NOPAR=0                 ;NO PARITY
001000                  ODDPAR=1000             ;ODD PARITY
001400                  EVEPAR=1400             ;EVEN PARITY
                                ;TXCSR BIT DEFINITIONS
100000                  DNA=BIT15               ;DATA NOT AVAILABLE
040000                  MTDATA=BIT14            ;MAINT DATA
020000                  CLK=BIT13               ;CLK
002000                  BITW=BIT10              ;BIT WINDOW
000400                  MRESET=BIT8             ;MASTER RESET
000200                  TXDONE=BIT7             ;XMIT DONE
000100                  TXINTE=BIT6             ;XMIT INTR ENABLE
```

```
806          000040                  DNAINTE=BIT5                ;DNA INTR ENAB
807          000020                  SEND=BIT4                   ;SEND
808          000010                  HDXEN=BIT3                  ;HDX/FDX
809          000001                  BREAK=BIT0                  ;BREAK
810                                          ;TXCSR WRD DEFINITIONS
811          000000                  USER=0                      ;USER MODE
812          004000                  MINT=4000                   ;MAINT INT MODE
813          010000                  MEXT=10000                  ;MAINT EXT MODE
814          014000                  SYSTST=14000                ;SYSTEM TEST MODE
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                        001400          .=.
001400                  $CMTAG:                         ;;START OF COMMON TAGS
001400  000000                  .WORD   0
001402  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001403  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001404  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001406  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001410  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001412  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001414  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001415  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001416  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001420  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001422  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001424  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001426  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001430  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001432  000000                  .WORD   0
001434  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001435  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001436  000000                  .WORD   0
001440  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001442  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001444  177560          $TKS:   177560                  ;;TTY KBD STATUS
001446  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001450  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG.  ADDRESS
001452  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG.  ADDRESS
001454  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001455  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001456  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS.  AFTER A "LINE FEED"
001457  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001460  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001462  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001464  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001466  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001470  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001472  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001474  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001476  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001500  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001502  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001504  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001506  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001510  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001512  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001514  000000          $ESCAPE: 0                      ;;ESCAPE ON ERROR ADDRESS
001516  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001522  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001523  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
```

```
                                        ;SLF:     .ASCIZ  <12>              ;;LINE FEED
                                        ;;*****************************************************************
                                        .SBTTL  APT MAILBOX-ETABLE

                                        ;;*****************************************************************
                                        .EVEN
001526                                  $MAIL:                              ;;APT MAILBOX
001526  000000                          $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
001530  000000                          $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
001532  000000                          $TESTN: .WORD   ATESTN   ;;TEST NUMBER
001534  000000                          $PASS:  .WORD   APASS    ;;PASS COUNT
001536  000000                          $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
001540  000000                          $UNIT:  .WORD   AUNIT    ;;I/O UNIT NUMBER
001542  000000                          $MSGAD: .WORD   AMSGAD   ;;MESSAGE ADDRESS
001544  000000                          $MSGLG: .WORD   AMSGLG   ;;MESSAGE LENGTH
001546                                  $ETABLE:                            ;;APT ENVIRONMENT TABLE
001546     000                          $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
001547     000                          $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
001550  000000                          $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
001552  000000                          $USWR:  .WORD   AUSWR    ;;USER SWITCHES
001554  000000                          $CPUOP: .WORD   ACPUOP   ;;CPU TYPE,OPTIONS
                                        ;*                                  BITS 15-11=CPU TYPE
                                        ;*                                          11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                        ;*                                          11/70=06,PDQ=07,Q=10
                                        ;*                                  BIT 10=REAL TIME CLOCK
                                        ;*                                  BIT  9=FLOATING POINT PROCESSOR
                                        ;*                                  BIT  8=MEMORY MANAGEMENT
001556     000                          $MAMS1: .BYTE   AMAMS1   ;;HIGH ADDRESS,M.S.  BYTE
001557     000                          $MTYP1: .BYTE   AMTYP1   ;;MEM.  TYPE,BLK#1
                                        ;*                                  MEM.TYPE BYTE   --  (HIGH BYTE)
                                        ;*                                          900 NSEC CORE=001
                                        ;*                                          300 NSEC BIPOLAR=002
                                        ;*                                          500 NSEC MOS=003
001560  000000                          $MADR1: .WORD   AMADR1   ;;HIGH ADDRESS,BLK#1
                                        ;*                                    MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001562     000                          $MAMS2: .BYTE   AMAMS2   ;;HIGH ADDRESS,M.S.  BYTE
001563     000                          $MTYP2: .BYTE   AMTYP2   ;;MEM.TYPE,BLK#2
001564  000000                          $MADR2: .WORD   AMADR2   ;,MEM.LAST ADDRESS,BLK#2
001566     000                          $MAMS3: .BYTE   AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
001567     000                          $MTYP3: .BYTE   AMTYP3   ;;MEM.TYPE,BLK#3
001570  000000                          $MADR3: .WORD   AMADR3   ;;MEM.LAST ADDRESS,BLK#3
001572     000                          $MAMS4: .BYTE   AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
001573     000                          $MTYP4: .BYTE   AMTYP4   ;;MEM.TYPE,BLK#4
001574  000000                          $MADR4: .WORD   AMADR4   ;;MEM.LAST ADDRESS,BLK#4
001576  000000                          $VECT1: .WORD   AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001600  000000                          $VECT2: .WORD   AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001602  000000                          $BASE:  .WORD   ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001604  000000                          $DEVM:  .WORD   ADEVM    ;;DEVICE MAP
001606  000000                          $CDW1:  .WORD   ACDW1    ;;CONTROLLER DESCRIPTION WORD#1
001610  000000                          $CDW2:  .WORD   ACDW2    ;;CONTROLLER DESCRIPTION WORD#2
001612  000000                          $DDW0:  .WORD   ADDW0    ;;DEVICE DESCRIPTOR WORD#0
001614  000000                          $DDW1:  .WORD   ADDW1    ;;DEVICE DESCRIPTOR WORD#1
001616  000000                          $DDW2:  .WORD   ADDW2    ;;DEVICE DESCRIPTOR WORD#2
001620  000000                          $DDW3:  .WORD   ADDW3    ;;DEVICE DESCRIPTOR WORD#3
001622  000000                          $DDW4:  .WORD   ADDW4    ;;DEVICE DESCRIPTOR WORD#4
001624  000000                          $DDW5:  .WORD   ADDW5    ;;DEVICE DESCRIPTOR WORD#5
```

```
   927  001626  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
   928  001630  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
   929  001632  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
   930  001634  000000                  $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
   931  001636  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
   932  001640  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
   933  001642  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
   934  001644  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
   935  001646  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
   936  001650  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
   937
   938
   939  001652                          $ETEND:
   940
   941
   942
```

```
                        ;INSTRUCTION DEFINITIONS

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                  PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                  POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                  POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                        ;REGISTER DEFINITIONS
                                ;RXCSR BIT DEFINITIONS
100000                  DSC=BIT15               ;DATA SET CHANGE
040000                  RING=BIT14              ;RING
020000                  CTS=BIT13               ;CLR TO SEND
010000                  CARDET=BIT12            ;CARRIER DETECT
004000                  RECACT=BIT11            ;REC ACTIVE
002000                  SRD=BIT10               ;SEC REC DATA
001000                  DSR=BIT9                ;DATA SET RDY
000400                  STPSYN=BIT8             ;STRIP SYNC
000200                  RXDONE=BIT7             ;REC DONE
000100                  RINTEN=BIT6             ;REC INTR ENABLE
000040                  DSINTE=BIT5             ;DSC INTR ENABLE
000020                  SYNSCH=BIT4             ;SYNC SEARCH
000010                  STD=BIT3                ;SEC XMIT DATA
000004                  RTS=BIT2                ;REQ TO SEND
000002                  DTR=BIT1                ;DATA TERM RDY
000001                  VOID=BIT0
                                ;RXDBUF BIT DEFINITIONS
100000                  RXERR=BIT15             ;REC ERROR
040000                  OVRRUN=BIT14            ;OVERRUN
020000                  FRMERR=BIT13            ;FRAME ERROR
010000                  PARER=BIT12             ;PARITY ERROR
                                ;PARCSR BIT DEFINITIONS
001000                  PAREN=BIT9              ;PARITY ENABLE
000400                  EVPAR=BIT8              ;EVEN PARITY SENSE
                                ;PARCSR WRD DEFINITIONS
030000                  SYNINT=30000            ;SYNC EXTERNAL MODE
020000                  SYNEXT=20000            ;SYNC INTERNAL MODE
000000                  ISYMOD=0                ;ISOC MODE
000000                  FIVE=0                  ;WORD LENGTH 5 BITS
002000                  SIX=2000                ;WORD LENGTH 6 BITS
004000                  SEVEN=4000              ;WORD LENGTH 7 BITS
006000                  EIGHT=6000              ;WORD LENGTH 8 BITS
000000                  NOPAR=0                 ;NO PARITY
001000                  ODDPAR=1000             ;ODD PARITY
001400                  EVEPAR=1400             ;EVEN PARITY
                                ;TXCSR BIT DEFINITIONS
100000                  DNA=BIT15               ;DATA NOT AVAILABLE
040000                  MTDATA=BIT14            ;MAINT DATA
020000                  CLK=BIT13               ;CLK
002000                  BITW=BIT10              ;BIT WINDOW
000400                  MRESET=BIT8             ;MASTER RESET
000200                  TXDONE=BIT7             ;XMIT DONE
000100                  TXINTE=BIT6             ;XMIT INTR ENABLE
```

```
  999          000040                  DNAINTE=BIT5                ;DNA INTR ENAB
 1000          000020                  SEND=BIT4                   ;SEND
 1001          000010                  HDXEN=BIT3                  ;HDX/FDX
 1002          000001                  BREAK=BIT0                  ;BREAK
 1003                                          ;TXCSR WRD DEFINITIONS
 1004          000000                  USER=0                      ;USER MODE
 1005          004000                  MINT=4000                   ;MAINT INT MODE
 1006          010000                  MEXT=10000                  ;MAINT EXT MODE
 1007          014000                  SYSTST=14000                ;SYSTEM TEST MODE
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*    . DF              ;;POINTS TO THE DATA FORMAT


001652                          $ERRTB:
                                        ;ERROR TABLE
001652  001762                          EM1     ;ERROR 1        REGISTER ERROR
001654  002067                          DH1
001656  002116                          DT1
001660  002132                          DF1
001662  002022                          EM2     ;ERROR 2        RECEIVER ERROR
001664  002067                          DH1
001666  002116                          DT1
001670  002132                          DF1
001672  002043                          EM3     ;ERROR 3        TRANSMITTER ERROR
001674  002067                          DH1
001676  002116                          DT1
001700  002132                          DF1
001702  001746                          EM4     ;ERROR 4        BIT ERROR (GENERAL)
001704  000000                          0
001706  002126                          DT4
001710  002132                          DF1

                                ;DEFAULT DU ADDRESSES
001712  160010                  RXCSR:  160010
001714  160011                  HRXCSR: 160011
001716  160012                  RXDBUF: 160012
001720  160013                  HRXDBUF:        160013
001722  160012                  PARCSR: 160012
001724  160013                  HPARCSR:        160013
001726  160014                  TXCSR:  160014
001730  160015                  HTXCSR: 160015
001732  160016                  TXDBUF: 160016
001734  160017                  HTXDBUF:        160017
                                ;DEFAULT DU VECTORS
001736  000770                  DURIV:  770     ;REC INTR VECTOR
001740  000772                  DURIS:  772     ;REC INTR STATUS
001742  000774                  DUTIV:  774     ;XMIT INTR VECTOR
001744  000776                  DUTIS:  776     ;XMIT INTR STATUS
                                ;ERROR MESSAGES
001746  020040  051105  047522  EM4:    .ASCIZ  /  ERROR PC /
001754  020122  041520  000040
001762  020040  047503  050115  EM1:    .ASCIZ /  COMPARISON ERROR ON REGISTERS/
001770  051101  051511  047117
001776  042440  051122  051117
002004  047440  020116  042522
```

```
1064  002012  044507  052123  051105
1065  002020  000123
1066  002022  020040  042522  042503  EM2:    .ASCIZ  /  RECEIVER ERROR/
1067  002030  053111  051105  042440
1068  002036  051122  051117     000
1069  002043     040  052040  040522  EM3:    .ASCIZ  /  TRANSMITTER ERROR/
1070  002050  051516  044515  052124
1071  002056  051105  042440  051122
1072  002064  051117     000
1073                                  ;DATA HEADERS FOR ERROR MESSAGES
1074  002067     105  051122  041520  DH1:    .ASCIZ  /ERRPC  WANTED  ACTUAL/
1075  002074  020040  040527  052116
1076  002102  042105  020040  041501
1077  002110  052524  046101     000
1078          002116                  .EVEN
1079                                  ;DATA TABLES FOR ERROR MESSAGES
1080  002116  001416  001130  001132  DT1:    .WORD   $ERRPC,HLD0,HLD1,0
1081  002124  000000
1082
1083  002126  001416  000000          DT4:    .WORD   $ERRPC,0
1084
1085  002132     000     000     000  DF1:    .BYTE   0,0,0,0
1086  002135     000
1087                                  .EVEN
1088                                  .SBTTL  ACT11 HOOKS
1089
1090                                  ;;*****************************************************************
1091                                  ;HOOKS REQUIRED BY ACT11
1092          002136                          $SVPC=.                 ;SAVE PC
1093          000046                          .=46
1094  000046  012660                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1095          000052                          .=52
1096  000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
1097          002136                          .=$SVPC                 ;; RESTORE PC
1098                                  .SBTTL  APT PARAMETER BLOCK
1099
1100                                  ;;*****************************************************************
1101                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1102                                  ;;*****************************************************************
1103          002136                          .$X=.   ;;SAVE CURRENT LOCATION
1104          000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1105  000024  000200                          200     ;;FOR APT START UP
1106          000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1107  000044  002136                          $APTHDR ;;POINT TO APT HEADER BLOCK
1108          002136                          .=.$X   ;;RESET LOCATION COUNTER
1109                                  ;;*****************************************************************
1110                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1111                                  ;INTERFACE SPEC.
1112
1113  002136                          $APTHD:
1114  002136  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1115  002140  001526                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1116  002142  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
1117  002144  000010                  $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1118  002146  000000                  $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1119  002150  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
1120
1121
1122                                    ;PROGRAM INITIALIZATION
1123                                    ;LOCK OUT INTERRUPTS
1124                                    ;SET UP PROCESSOR STACK
1125                                    ;SET UP POWER FAIL VECTOR
1126                                    ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1127                                    ;TYPE TITLE MESSAGE
1128
1129  002152                        .START:
1130                                .SBTTL  INITIALIZE THE COMMON TAGS
1131                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1132  002152  012706  001400               MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1133  002156  005026                       CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1134  002160  022706  001440               CMP     #SWR,R6 ;;DONE?
1135  002164  001374                       BNE     .-6             ;;LOOP BACK IF NO
1136  002166  012706  001100               MOV     #@#STACK,SP             ;;SETUP THE STACK POINTER
1137                                ;;INITIALIZE A FEW VECTORS
1138  002172  012737  016304  000020       MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1139  002200  012737  000340  000022       MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1140  002206  012737  014174  000030       MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1141  002214  012737  000340  000032       MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1142  002222  012737  016640  000034       MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1143  002230  012737  000340  000036       MOV     #340,@#TRAPVEC+2;LEVEL 7
1144  002236  012737  014776  000024       MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1145  002244  012737  000340  000026       MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1146  002252  005067  177234               CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1147  002256  005067  177232               CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1148  002262  112767  000001  177125       MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1149  002270  012767  002270  177110       MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1150  002276  012767  002276  177104       MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1151                                ;;SIZE FOR A HARDWARE SWITCH REGISTER.  IF NOT FOUND OR IT IS
1152                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1153  002304  013746  000004               MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1154  002310  012737  002344  000004       MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1155  002316  012767  177570  177114       MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1156  002324  012767  177570  177110       MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1157  002332  022777  177777  177100       CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1158  002340  001012                       BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1159                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
1160  002342  000403                       BR      65$             ;;BRANCH IF NO TIMEOUT
1161  002344  012716  002352       64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1162  002350  000002                       RTI
1163  002352  012767  000176  177060 65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1164  002360  012767  000174  177054       MOV     #DISPREG,DISPLAY
1165  002366  012637  000004       66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1166
1167  002372  005067  177136               CLR     $PASS           ;;CLEAR PASS COUNT
1168  002376  132767  000200  177143       BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1169  002404  001403                       BEQ     67$             ;;YES,USE NON-APT SWITCH
1170  002406  012767  001550  177024       MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1171  002414                        67$:
1172  002414  012706  001100               MOV     #STACK,SP               ;SET STACK
1173  002420  106427  000340               MTPS    #340                    ;LOCK INTERRUPTS
1174  002424  012737  014776  000024       MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
1175  002432  105067  176535               CLRB    STFLG                   ;CLEAR START FLAG
```

```
1176  002436  005067  176450                CLR     PASCNT                  ;CLEAR PASS COUNT
1177  002442  105067  176735                CLRB    $ERFLG                  ;CLEAR ERROR FLAG
1178  002446  005067  176740                CLR     $ERTTL                  ;CLEAR ERROR COUNT
1179  002452  005067  176740                CLR     $ERRPC                  ;CLEAR LAST ERROR POINTER
1180  002456  012767  000001  176716        MOV     #1,$TSTNM               ;SET UP FOR TEST 1
1181  002464  012767  002152  176412        MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
1182                                                                        ;TESTING STARTS
1183  002472  013746  000006                MOV     @#6,-(SP)
1184  002476  013746  000004                MOV     @#4,-(SP)
1185  002502  012737  002516  000004        MOV     #1$,@#4
1186  002510  005777  176724                TST     @SWR
1187  002514  000407                        BR      2$
1188  002516  012767  000176  176714  1$:   MOV     #SWREG,SWR
1189  002524  012767  000174  176710        MOV     #DISPREG,DISPLAY
1190  002532  022626                        CMP     (SP)+,(SP)+
1191  002534  012637  000004          2$:   MOV     (SP)+,@#4
1192  002540  012637  000006                MOV     (SP)+,@#6
1193  002544  022767  000176  176666        CMP     #SWREG,SWR
1194  002552  001007                        BNE     3$
1195  002554  005737  000042                TST     @#42                    ;CHECK FOR CHAIN
1196  002560  001402                        BEQ     33$
1197  002562  000167  000522                JMP     .BEGIN
1198  002566  004767  010170          33$:  JSR     PC,CNTLU
1199  002572  105767  176374          3$:   TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
1200  002576  001004                        BNE     ONCE
1201  002600  104401  015136                TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
1202  002604  105167  176362                COMB    INIFLG                  ;IF NOT SET FLAG AND DO
1203  002610  105767  176732          ONCE: TSTB    $ENV            ;APT CONTROL?
1204  002614  001410                        BEQ     11$             ;BR IF NO
1205  002616  032767  000001  176726        BIT     #1,$USWR        ;EXTENAL JUMPER ON?
1206  002624  001002                        BNE     12$             ;NO
1207  002626  105067  176321                CLRB    JMRBY           ;CLEAR FLAG
1208  002632  000167  000452          12$:  JMP     .BEGIN          ;GO DO IT
1209  002636  032777  000001  176574  11$:  BIT     #SW00,@SWR      ;RESELECT VECTOR & CONTROL REG?
1210  002644  001002                        BNE     1$
1211  002646  000167  000436                JMP     .BEGIN
1212  002652  012700  000300          1$:   MOV     #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1213  002656  012701  000302                MOV     #302,R1 ;START AT LOCATION 300
1214  002662  012702  000004                MOV     #4,R2
1215  002666  010110                  2$:   MOV     R1,(R0)
1216  002670  005011                        CLR     (R1)
1217  002672  060200                        ADD     R2,R0
1218  002674  060201                        ADD     R2,R1
1219  002676  022701  001000                CMP     #1000,R1        ;END AT LOCATION 776
1220  002702  002771                        BLT     2$
1221  002704  104406                        INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1222  002706  015204                        MREGAD  ;MESSAGE
1223  002710  104410                        PARAM   ;CONVERT STRING
1224  002712  160000                        160000  ;LOW LIMIT
1225  002714  167776                        167776  ;HIGH LIMIT
1226  002716  017134                        DUBASE  ;STORE AT THIS LOCATION
1227  002720     001                  .BYTE  1       ;MASK
1228  002721     001                  .BYTE  1       ;HOW MANY TIMES + 2
1229  002722  016767  014206  176226        MOV     DUBASE,KEEPADD  ;SAVE
1230  002730  004767  014046                JSR     PC,DUADDR
1231  002734  016767  176216  176212        MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
```

```
1232  002742  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1233  002744  015171                          MVECTO  ;MESSAGE
1234  002746  104410                          PARAM   ;CONVERT STRING
1235  002750  000300                          300     ;LOW LIMIT
1236  002752  000776                          776     ;HIGH LIMIT
1237  002754  001736                          DURIV   ;STORE AT THIS LOCATION
1238  002756     001                  .BYTE   1       ;MASK
1239  002757     004                  .BYTE   4       ;HOW MANY TIMES + 2
1240  002760  016767  176752  176176          MOV     DURIV,KEEPIV    ;SAVE
1241  002766  016767  176744  176166          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
1242  002774  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1243  002776  015234                          MMULT   ;MESSAGE
1244  003000  104414                          SETFLG  ;SET FLAG BASED UPON INPUT STRING
1245  003002  001152                          MULTD   ;THIS FLAG
1246  003004  105767  176142                  TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
1247                                                          ;ON THE SYSTEM ?
1248  003010  100406                          BMI     BBB     ;YES, ASK NEXT QUESTION
1249  003012  005067  176150                  CLR     ACTREG
1250  003016  005067  176146                  CLR     ROTADD
1251  003022  000167  000140                  JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
1252  003026                          BBB:
1253  003026  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1254  003030  015263                          MLASTD  ;MESSAGE
1255  003032  104410                          PARAM   ;CONVERT STRING
1256  003034  160000                          160000  ;LOW LIMIT
1257  003036  167776                          167776  ;HIGH LIMIT
1258  003040  001160                          LASTADD ;STORE AT THIS LOCATION
1259  003042     001                  .BYTE   1       ;MASK
1260  003043     001                  .BYTE   1       ;HOW MANY TIMES + 2
1261                                  ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1262  003044  012767  000001  176116  1$:     MOV     #1,ROTADD       ;SET UP POINTER
1263  003052  005067  176110                  CLR     ACTREG  ;CLR ACTIVE REGISTER
1264  003056  056767  176106  176102  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
1265  003064  000241                          CLC
1266  003066  006167  176076                  ROL     ROTADD  ;SET UP POINTER
1267  003072  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
1268  003074  062767  000010  176052          ADD     #10,BASEADD     ;SET UP BASE ADDRESS
1269  003102  026767  176052  176044          CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1270  003110  101362                          BHI     2$      ;NO DO IT AGAIN
1271  003112  056767  176052  176046          BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
1272                                                          ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1273                                                          ;MULTIPLE DEVICE QUESTION
1274  003120  012767  000001  176042  4$:     MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1275  003126  016767  176024  176020          MOV     KEEPADD,BASEADD ;DITTO
1276  003134  000414                          BR      OUTMUL  ;CONTINUE QUESTIONS
1277  003136  016767  176014  176010  3$:     MOV     KEEPADD,BASEADD ;RESTORE
1278  003144  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1279  003146  015357                          MRANGE  ;MESSAGE
1280  003150  104410                          PARAM   ;CONVERT STRING
1281  003152  160000                          160000  ;LOW LIMIT
1282  003154  167776                          167776  ;HIGH LIMIT
1283  003156  001160                          LASTADD ;STORE AT THIS LOCATION
1284  003160     001                  .BYTE   1       ;MASK
1285  003161     001                  .BYTE   1       ;HOW MANY TIMES + 2
1286  003162  000167  177656                  JMP     1$      ;DO IT AGAIN
1287  003166  012767  000340  013602  OUTMUL: MOV     #340,DUPRT
```

```
1288  003174  004767  013526              JSR     PC,DULEV
1289                                      ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1290                                      ;BUFFER TO THE CHARACTERS "1" AND "2".
1291                                      ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1292                                      ;IF THE CHARACTER IS "2" SET THE FLAG
1293  003200                      AAA:
1294  003200  104406                     INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1295  003202  015575                     MSYNC   ;MESSAGE
1296  003204  122767  000061  012724  3$: CMPB    #'1,INBUF       ;IS IT "1" ?
1297  003212  001003                     BNE     1$
1298  003214  105067  175726            CLRB    SYNCNO  ;000
1299  003220  000412                     BR      4$
1300  003222  122767  000062  012706  1$: CMPB    #'2,INBUF       ;IS IT "2" ?
1301  003230  001004                     BNE     2$
1302  003232  112767  177777  175706        MOVB    #-1,SYNCNO      ;377
1303  003240  000402                     BR      4$
1304  003242  104407                 2$: INSTER  ;RETRY
1305  003244  000757                     BR      3$
1306  003246  000240                 4$: NOP
1307  003250  104406                     INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1308  003252  015643                     MWIRE6  ;MESSAGE
1309  003254  104414                     SETFLG  ;SET FLAG BASED UPON INPUT STRING
1310  003256  001147                     SEXMIT  ;THIS FLAG
1311  003260  104406                     INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1312  003262  015714                     MWIRE5  ;MESSAGE
1313  003264  104414                     SETFLG  ;SET FLAG BASED UPON INPUT STRING
1314  003266  001150                     SEREC   ;THIS FLAG
1315  003270  104406                     INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1316  003272  015764                     MWIRE4  ;MESSAGE
1317  003274  104414                     SETFLG  ;SET FLAG BASED UPON INPUT STRING
1318  003276  001151                     OPTCLR  ;THIS FLAG
1319  003300  104406                     INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1320  003302  016043                     MEXTJ   ;MESSAGE
1321  003304  104414                     SETFLG  ;SET FLAG BASED UPON INPUT STRING
1322  003306  001153                     JMRBY   ;THIS FLAG
1323
1324                                      ;TEST START AND RESTART
1325
1326  003310  012706  001100      .BEGIN: MOV     #STACK,SP               ;SET UP STACK
1327  003314  106427  000340             MTPS    #340                    ;LOCK OUT INTERRUPTS
1328  003320  032777  000002  176112     BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
1329  003326  001413                     BEQ     3$
1330  003330  104406                     INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1331  003332  015527                     MTSTPC  ;MESSAGE
1332  003334  104410                     PARAM   ;CONVERT STRING
1333  003336  003374                     TST1    ;LOW LIMIT
1334  003340  017500                     17500   ;HIGH LIMIT
1335  003342  001402                     $TSTNM  ;STORE AT THIS LOCATION
1336  003344     001              .BYTE   1       ;MASK
1337  003345     001              .BYTE   1       ;HOW MANY TIMES + 2
1338  003346  016767  176030  175530     MOV     $TSTNM,RETURN
1339  003354  000403                     BR      4$
1340  003356  012767  003374  175520  3$: MOV     #TST1,RETURN            ;START AT TEST 1
1341  003364  104401  015523          4$: TYPE    ,MR                     ;TYPE R
1342  003370  000177  175510             JMP     @RETURN                 ;START TESTING
1343
```

```
1344                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1345                                    ;;RECEIVER SECTION, IT USES THE ERROR FLAGS
1346                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1347                                    ;;(OVRRUN,RXERR)
1348                                    ;;MODE: SYNEXT
1349                                    ;;LENGTH: SEVEN
1350                                    ;;CHAR: 177
1351                                    ;;
1352                                ;;*****************************************************************
1353  003374  000004                TST1:   SCOPE
1354  003376  052777  000400  176322        BIS     #MRESET,@TXCSR  ;MASTER RESET
1355  003404  012777  020000  176310        MOV     #SYNEXT,@PARCSR ;SET THE MODE
1356  003412  052777  000400  176306        BIS     #MRESET,@TXCSR  ;MASTER RESET
1357
1358                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1359  003420  012777  064001  176300        MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1360
1361                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1362  003426  012777  024000  176266        MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
1363  003434  052777  000020  176250        BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1364                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1365  003442  042777  020000  176256        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1366  003450  052777  020000  176250        BIS     #CLK,@TXCSR     ;POKE CLK UP
1367  003456  016703  176234              MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1368  003462  012700  000177              MOV     #177,R0 ;EXPECTED
1369  003466  012767  000007  175426      MOV     #7,SHIFT        ;# OF SHIFTS
1370  003474  012767  000177  175776      MOV     #177,$TMP1      ;DATA CHAR
1371  003502  004767  013430              JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1372  003506  105777  176200              TSTB    @RXCSR  ;RXDONE ?
1373  003512  100401                      BMI     .+4
1374  003514  104004                      ERROR   4       ;RXDONE SHOULD BE SET
1375  003516  017701  176174              MOV     @RXDBUF,R1      ;ACTUAL
1376  003522  020001                      CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1377  003524  001401                      BEQ     .+4
1378  003526  104002                      ERROR   2       ;RECEIVED DATA DID NOT MATCH
1379                                                      ;EXPECTED DATA - CHECK MAINT DATA
1380                                                      ;OR RECEIVER LOGIC
1381  003530  012767  000007  175364      MOV     #7,SHIFT        ;# OF SHIFTS
1382  003536  012767  000177  175734      MOV     #177,$TMP1      ;DATA CHAR
1383  003544  004767  013366              JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1384                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1385  003550  012767  000007  175344      MOV     #7,SHIFT        ;# OF SHIFTS
1386  003556  012767  000177  175714      MOV     #177,$TMP1      ;DATA CHAR
1387  003564  004767  013346              JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1388  003570  012700  140177              MOV     #140000!177,R0  ;EXPECTED DATA PLUS
1389                                                              ;RXERR & OVRRUN
1390  003574  017701  176116              MOV     @RXDBUF,R1      ;ACTUAL
1391  003600  020001                      CMP     R0,R1   ;COMPARE EXP VS. ACT
1392  003602  001401                      BEQ     .+4
1393  003604  104002                      ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1394                                                      ;OVRRUN BITS... THEY BOTH SHOULD BE SET
1395
1396                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1397                                    ;;RECEIVER SECTION, IT USES THE ERROR FLAGS
1398                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1399                                    ;;(OVRRUN,RXERR)
```

```
1400                                            ;;MODE: SYNEXT
1401                                            ;;LENGTH: SEVEN
1402                                            ;;CHAR: 0
1403                                            ;;
1404                                    ;;**************************************************************************
1405  003606  000004                    TST2:   SCOPE
1406  003610  052777  000400  176110            BIS     #MRESET,@TXCSR  ;MASTER RESET
1407  003616  012777  020000  176076            MOV     #SYNEXT,@PARCSR ;SET THE MODE
1408  003624  052777  000400  176074            BIS     #MRESET,@TXCSR  ;MASTER RESET
1409
1410                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1411  003632  012777  064001  176066            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1412
1413                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1414  003640  012777  024000  176054            MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
1415  003646  052777  000020  176036            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1416                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1417  003654  042777  020000  176044            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1418  003662  052777  020000  176036            BIS     #CLK,@TXCSR     ;POKE CLK UP
1419  003670  016703  176022                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1420  003674  012700  000000                    MOV     #0,R0   ;EXPECTED
1421  003700  012767  000007  175214            MOV     #7,SHIFT        ;# OF SHIFTS
1422  003706  012767  000000  175564            MOV     #0,$TMP1        ;DATA CHAR
1423  003714  004767  013216                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1424  003720  105777  175766                    TSTB    @RXCSR  ;RXDONE ?
1425  003724  100401                            BMI     .+4
1426  003726  104004                            ERROR   4       ;RXDONE SHOULD BE SET
1427  003730  017701  175762                    MOV     @RXDBUF,R1      ;ACTUAL
1428  003734  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1429  003736  001401                            BEQ     .+4
1430  003740  104002                            ERROR   2       ;RECEIVED DATA DID NOT MATCH
1431                                                            ;EXPECTED DATA - CHECK MAINT DATA
1432                                                            ;OR RECEIVER LOGIC
1433  003742  012767  000007  175152            MOV     #7,SHIFT        ;# OF SHIFTS
1434  003750  012767  000000  175522            MOV     #0,$TMP1        ;DATA CHAR
1435  003756  004767  013154                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1436                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1437  003762  012767  000007  175132            MOV     #7,SHIFT        ;# OF SHIFTS
1438  003770  012767  000000  175502            MOV     #0,$TMP1        ;DATA CHAR
1439  003776  004767  013134                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1440  004002  012700  140000                    MOV     #140000!0,R0    ;EXPECTED DATA PLUS
1441                                                            ;RXERR & OVRRUN
1442  004006  017701  175704                    MOV     @RXDBUF,R1      ;ACTUAL
1443  004012  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1444  004014  001401                            BEQ     .+4
1445  004016  104002                            ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1446                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1447
1448                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1449                                            ;;RECEIVER SECTION, IT USES THE ERROR FLAGS
1450                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1451                                            ;;(OVRRUN,RXERR)
1452                                            ;;MODE: SYNEXT
1453                                            ;;LENGTH: EIGHT
1454                                            ;;CHAR: 125
1455                                            ;;
```

```
1456                                    ;;**************************************************************
1457  004020  000004                    TST3:   SCOPE
1458  004022  052777  000400  175676            BIS     #MRESET,@TXCSR  ;MASTER RESET
1459  004030  012777  020000  175664            MOV     #SYNEXT,@PARCSR ;SET THE MODE
1460  004036  052777  000400  175662            BIS     #MRESET,@TXCSR  ;MASTER RESET
1461
1462                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1463  004044  012777  064001  175654            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1464
1465                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1466  004052  012777  026000  175642            MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1467  004060  052777  000020  175624            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1468                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1469  004066  042777  020000  175632            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1470  004074  052777  020000  175624            BIS     #CLK,@TXCSR     ;POKE CLK UP
1471  004102  016703  175610                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1472  004106  012700  000125                    MOV     #125,R0 ;EXPECTED
1473  004112  012767  000010  175002            MOV     #8.,SHIFT       ;# OF SHIFTS
1474  004120  012767  000125  175352            MOV     #125,$TMP1      ;DATA CHAR
1475  004126  004767  013004                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1476  004132  105777  175554                    TSTB    @RXCSR  ;RXDONE ?
1477  004136  100401                            BMI     .+4
1478  004140  104004                            ERROR   4       ;RXDONE SHOULD BE SET
1479  004142  017701  175550                    MOV     @RXDBUF,R1      ;ACTUAL
1480  004146  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1481  004150  001401                            BEQ     .+4
1482  004152  104002                            ERROR   2       ;RECEIVED DATA DID NOT MATCH
1483                                                            ;EXPECTED DATA - CHECK MAINT DATA
1484                                                            ;OR RECEIVER LOGIC
1485  004154  012767  000010  174740            MOV     #8.,SHIFT       ;# OF SHIFTS
1486  004162  012767  000125  175310            MOV     #125,$TMP1      ;DATA CHAR
1487  004170  004767  012742                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1488                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1489  004174  012767  000010  174720            MOV     #8.,SHIFT       ;# OF SHIFTS
1490  004202  012767  000125  175270            MOV     #125,$TMP1      ;DATA CHAR
1491  004210  004767  012722                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1492  004214  012700  140125                    MOV     #140000!125,R0  ;EXPECTED DATA PLUS
1493                                                            ;RXERR & OVRRUN
1494  004220  017701  175472                    MOV     @RXDBUF,R1      ;ACTUAL
1495  004224  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1496  004226  001401                            BEQ     .+4
1497  004230  104002                            ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1498                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1499
1500                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1501                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1502                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1503                                            ;;(OVRRUN,RXERR)
1504                                            ;;MODE: SYNEXT
1505                                            ;;LENGTH: EIGHT
1506                                            ;;CHAR: 252
1507                                            ;;
1508                                    ;;**************************************************************
1509  004232  000004                    TST4:   SCOPE
1510  004234  052777  000400  175464            BIS     #MRESET,@TXCSR  ;MASTER RESET
1511  004242  012777  020000  175452            MOV     #SYNEXT,@PARCSR ;SET THE MODE
```

```
1512  004250  052777  000400  175450          BIS     #MRESET,@TXCSR  ;MASTER RESET
1513
1514                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1515  004256  012777  064001  175442          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1516
1517                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1518  004264  012777  026000  175430          MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1519  004272  052777  000020  175412          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1520                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1521  004300  042777  020000  175420          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1522  004306  052777  020000  175412          BIS     #CLK,@TXCSR     ;POKE CLK UP
1523  004314  016703  175376                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1524  004320  012700  000252                  MOV     #252,R0 ;EXPECTED
1525  004324  012767  000010  174570          MOV     #8.,SHIFT       ;# OF SHIFTS
1526  004332  012767  000252  175140          MOV     #252,$TMP1      ;DATA CHAR
1527  004340  004767  012572                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1528  004344  105777  175342                  TSTB    @RXCSR  ;RXDONE ?
1529  004350  100401                          BMI     .+4
1530  004352  104004                          ERROR   4       ;RXDONE SHOULD BE SET
1531  004354  017701  175336                  MOV     @RXDBUF,R1      ;ACTUAL
1532  004360  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1533  004362  001401                          BEQ     .+4
1534  004364  104002                          ERROR   2       ;RECEIVED DATA DID NOT MATCH
1535                                                          ;EXPECTED DATA - CHECK MAINT DATA
1536                                                          ;OR RECEIVER LOGIC
1537  004366  012767  000010  174526          MOV     #8.,SHIFT       ;# OF SHIFTS
1538  004374  012767  000252  175076          MOV     #252,$TMP1      ;DATA CHAR
1539  004402  004767  012530                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1540                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1541  004406  012767  000010  174506          MOV     #8.,SHIFT       ;# OF SHIFTS
1542  004414  012767  000252  175056          MOV     #252,$TMP1      ;DATA CHAR
1543  004422  004767  012510                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1544  004426  012700  140252                  MOV     #140000!252,R0  ;EXPECTED DATA PLUS
1545                                                          ;RXERR & OVRRUN
1546  004432  017701  175260                  MOV     @RXDBUF,R1      ;ACTUAL
1547  004436  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1548  004440  001401                          BEQ     .+4
1549  004442  104002                          ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1550                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1551
1552                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1553                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1554                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1555                                          ;;(OVRRUN,RXERR)
1556                                          ;;MODE: SYNEXT
1557                                          ;;LENGTH: EIGHT
1558                                          ;;CHAR: 377
1559                                          ;;
1560                                  ;;**************************************************************************
1561  004444  000004                  TST5:   SCOPE
1562  004446  052777  000400  175252          BIS     #MRESET,@TXCSR  ;MASTER RESET
1563  004454  012777  020000  175240          MOV     #SYNEXT,@PARCSR ;SET THE MODE
1564  004462  052777  000400  175236          BIS     #MRESET,@TXCSR  ;MASTER RESET
1565
1566                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1567  004470  012777  064001  175230          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
```

```
1568
1569                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1570  004476  012777  026000  175216            MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1571  004504  052777  000020  175200            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1572                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1573  004512  042777  020000  175206            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1574  004520  052777  020000  175200            BIS     #CLK,@TXCSR     ;POKE CLK UP
1575  004526  016703  175164                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1576  004532  012700  000377                    MOV     #377,R0 ;EXPECTED
1577  004536  012767  000010  174356            MOV     #8.,SHIFT       ;# OF SHIFTS
1578  004544  012767  000377  174726            MOV     #377,$TMP1      ;DATA CHAR
1579  004552  004767  012360                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1580  004556  105777  175130                    TSTB    @RXCSR  ;RXDONE ?
1581  004562  100401                            BMI     .+4
1582  004564  104004                            ERROR   4       ;RXDONE SHOULD BE SET
1583  004566  017701  175124                    MOV     @RXDBUF,R1      ;ACTUAL
1584  004572  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1585  004574  001401                            BEQ     .+4
1586  004576  104002                            ERROR   2       ;RECEIVED DATA DID NOT MATCH
1587                                                            ;EXPECTED DATA - CHECK MAINT DATA
1588                                                            ;OR RECEIVER LOGIC
1589  004600  012767  000010  174314            MOV     #8.,SHIFT       ;# OF SHIFTS
1590  004606  012767  000377  174664            MOV     #377,$TMP1      ;DATA CHAR
1591  004614  004767  012316                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1592                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1593  004620  012767  000010  174274            MOV     #8.,SHIFT       ;# OF SHIFTS
1594  004626  012767  000377  174644            MOV     #377,$TMP1      ;DATA CHAR
1595  004634  004767  012276                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1596  004640  012700  140377                    MOV     #140000!377,R0  ;EXPECTED DATA PLUS
1597                                                            ;RXERR & OVRRUN
1598  004644  017701  175046                    MOV     @RXDBUF,R1      ;ACTUAL
1599  004650  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1600  004652  001401                            BEQ     .+4
1601  004654  104002                            ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1602                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1603
1604                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1605                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1606                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1607                                            ;;(OVRRUN,RXERR)
1608                                            ;;MODE: SYNEXT
1609                                            ;;LENGTH: EIGHT
1610                                            ;;CHAR: 0
1611                                            ;;
1612                                    ;;**************************************************************************
1613  004656  000004                    TST6:   SCOPE
1614  004660  052777  000400  175040            BIS     #MRESET,@TXCSR  ;MASTER RESET
1615  004666  012777  020000  175026            MOV     #SYNEXT,@PARCSR ;SET THE MODE
1616  004674  052777  000400  175024            BIS     #MRESET,@TXCSR  ;MASTER RESET
1617
1618                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1619  004702  012777  064001  175016            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1620
1621                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1622  004710  012777  026000  175004            MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1623  004716  052777  000020  174766            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
```

```
1624                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1625  004724  042777  020000  174774            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1626  004732  052777  020000  174766            BIS     #CLK,@TXCSR     ;POKE CLK UP
1627  004740  016703  174752                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1628  004744  012700  000000                    MOV     #0,R0   ;EXPECTED
1629  004750  012767  000010  174144            MOV     #8.,SHIFT       ;# OF SHIFTS
1630  004756  012767  000000  174514            MOV     #0,$TMP1        ;DATA CHAR
1631  004764  004767  012146                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1632  004770  105777  174716                    TSTB    @RXCSR  ;RXDONE ?
1633  004774  100401                            BMI     .+4
1634  004776  104004                            ERROR   4       ;RXDONE SHOULD BE SET
1635  005000  017701  174712                    MOV     @RXDBUF,R1      ;ACTUAL
1636  005004  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1637  005006  001401                            BEQ     .+4
1638  005010  104002                            ERROR   2       ;RECEIVED DATA DID NOT MATCH
1639                                                            ;EXPECTED DATA - CHECK MAINT DATA
1640                                                            ;OR RECEIVER LOGIC
1641  005012  012767  000010  174102            MOV     #8.,SHIFT       ;# OF SHIFTS
1642  005020  012767  000000  174452            MOV     #0,$TMP1        ;DATA CHAR
1643  005026  004767  012104                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1644                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1645  005032  012767  000010  174062            MOV     #8.,SHIFT       ;# OF SHIFTS
1646  005040  012767  000000  174432            MOV     #0,$TMP1        ;DATA CHAR
1647  005046  004767  012064                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1648  005052  012700  140000                    MOV     #140000!0,R0    ;EXPECTED DATA PLUS
1649                                                            ;RXERR & OVRRUN
1650  005056  017701  174634                    MOV     @RXDBUF,R1      ;ACTUAL
1651  005062  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1652  005064  001401                            BEQ     .+4
1653  005066  104002                            ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1654                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1655
1656                                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1657                                                    ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1658                                                    ;;THAT IT WAS SELECTED PROPERLY
1659                                                    ;;FRAME ERROR (FRMERR,RXERR)
1660                                                    ;;MODE: ISOC (ISYMOD)
1661                                                    ;;LENGTH: FIVE
1662                                                    ;;CHAR:  25
1663                                                    ;;
1664                                    ;;**************************************************************************
1665  005070  000004                    TST7:   SCOPE
1666  005072  052777  000400  174626            BIS     #MRESET,@TXCSR  ;MASTER RESET
1667  005100  012777  000000  174614            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1668  005106  052777  000400  174612            BIS     #MRESET,@TXCSR  ;MASTER RESET
1669
1670                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1671  005114  012777  064001  174604            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1672
1673                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1674  005122  012777  000000  174572            MOV     #ISYMOD!FIVE!NOPAR!0,@PARCSR
1675  005130  052777  000020  174554            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1676                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION...
1677  005136  042777  020000  174562            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1678  005144  052777  020000  174554            BIS     #CLK,@TXCSR     ;POKE CLK UP
1679                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
```

```
1680  005152  042777  020000  174546          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1681  005160  052777  020000  174540          BIS     #CLK,@TXCSR     ;POKE CLK UP
1682  005166  012767  000007  173726          MOV     #7,SHIFT        ;# OF SHIFTS
1683  005174  012767  000052  174276          MOV     #52,$TMP1       ;DATA CHAR
1684                                  ;NOTE:  THE ABOVE CHARACTER IS MISSING STOP BIT
1685  005202  004767  011730                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1686  005206  016703  174504                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1687  005212  012700  120025                  MOV     #RXERR!FRMERR!25,R0     ;EXPECTED
1688  005216  017701  174474                  MOV     @RXDBUF,R1      ;ACTUAL
1689  005222  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1690  005224  001401                          BEQ     .+4
1691  005226  104000                          ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
1692                                                  ;IF LOWER BYTE DOES NOT MATCH  IT
1693                                                  ;PROBABLY IS A LENGTH SELECT PROBLEM
1694
1695                                  ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1696                                  ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1697                                  ;;THAT IT WAS SELECTED PROPERLY
1698                                  ;;FRAME ERROR (FRMERR,RXERR)
1699                                  ;;MODE: ISOC (ISYMOD)
1700                                  ;;LENGTH: SIX
1701                                  ;;CHAR:  25
1702                                  ;;
1703                          ;;*****************************************************************
1704  005230  000004          TST10:  SCOPE
1705  005232  052777  000400  174466          BIS     #MRESET,@TXCSR  ;MASTER RESET
1706  005240  012777  000000  174454          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1707  005246  052777  000400  174452          BIS     #MRESET,@TXCSR  ;MASTER RESET
1708
1709                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1710  005254  012777  064001  174444          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1711
1712                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1713  005262  012777  002000  174432          MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
1714  005270  052777  000020  174414          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1715                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1716  005276  042777  020000  174422          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1717  005304  052777  020000  174414          BIS     #CLK,@TXCSR     ;POKE CLK UP
1718                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1719  005312  042777  020000  174406          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1720  005320  052777  020000  174400          BIS     #CLK,@TXCSR     ;POKE CLK UP
1721  005326  012767  000010  173566          MOV     #8.,SHIFT       ;# OF SHIFTS
1722  005334  012767  000052  174136          MOV     #52,$TMP1       ;DATA CHAR
1723                                  ;NOTE:  THE ABOVE CHARACTER IS MISSING STOP BIT
1724  005342  004767  011570                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1725  005346  016703  174344                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1726  005352  012700  120025                  MOV     #RXERR!FRMERR!25,R0     ;EXPECTED
1727  005356  017701  174334                  MOV     @RXDBUF,R1      ;ACTUAL
1728  005362  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1729  005364  001401                          BEQ     .+4
1730  005366  104000                          ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
1731                                                  ;IF LOWER BYTE DOES NOT MATCH  IT
1732                                                  ;PROBABLY IS A LENGTH SELECT PROBLEM
1733
1734                                  ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1735                                  ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
```

```
 1736                                           ;;THAT IT WAS SELECTED PROPERLY
 1737                                           ;;FRAME ERROR (FRMERR,RXERR)
 1738                                           ;;MODE: ISOC (ISYMOD)
 1739                                           ;;LENGTH: SEVEN
 1740                                           ;;CHAR:  125
 1741                                           ;;
 1742                                   ;;**************************************************************************
 1743  005370  000004                  TST11:   SCOPE
 1744  005372  052777  000400  174326           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1745  005400  012777  000000  174314           MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1746  005406  052777  000400  174312           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1747
 1748                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1749  005414  012777  064001  174304           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1750
 1751                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1752  005422  012777  004000  174272           MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
 1753  005430  052777  000020  174254           BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1754                                           ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1755  005436  042777  020000  174262           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1756  005444  052777  020000  174254           BIS     #CLK,@TXCSR     ;POKE CLK UP
 1757                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1758  005452  042777  020000  174246           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1759  005460  052777  020000  174240           BIS     #CLK,@TXCSR     ;POKE CLK UP
 1760  005466  012767  000011  173426           MOV     #9.,SHIFT       ;# OF SHIFTS
 1761  005474  012767  000252  173776           MOV     #252,$TMP1      ;DATA CHAR
 1762                                   ;NOTE:  THE ABOVE CHARACTER IS MISSING STOP BIT
 1763  005502  004767  011430                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1764  005506  016703  174204                    MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
 1765  005512  012700  120125                    MOV     #RXERR!FRMERR!125,R0    ;EXPECTED
 1766  005516  017701  174174                    MOV     @RXDBUF,R1      ;ACTUAL
 1767  005522  020001                           CMP     R0,R1   ;COMPARE EXP VS ACT
 1768  005524  001401                           BEQ     .+4
 1769  005526  104000                           ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
                                                        ;IF LOWER BYTE DOES NOT MATCH  IT
                                                        ;PROBABLY IS A LENGTH SELECT PROBLEM
 1770
 1771
 1772
 1773                                           ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
 1774                                           ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
 1775                                           ;;THAT IT WAS SELECTED PROPERLY
 1776                                           ;;FRAME ERROR (FRMERR,RXERR)
 1777                                           ;;MODE: ISOC (ISYMOD)
 1778                                           ;;LENGTH: EIGHT
 1779                                           ;;CHAR:  125
 1780                                           ;;
 1781                                   ;;**************************************************************************
 1782  005530  000004                  TST12:   SCOPE
 1783  005532  052777  000400  174166           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1784  005540  012777  000000  174154           MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1785  005546  052777  000400  174152           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1786
 1787                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1788  005554  012777  064001  174144           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1789
 1790                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1791  005562  012777  006000  174132           MOV     #ISYMOD!EIGHT!NOPAR!0,@PARCSR
```

```
 1792  005570  052777  000020  174114          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1793                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1794  005576  042777  020000  174122          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1795  005604  052777  020000  174114          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1796                                      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1797  005612  042777  020000  174106          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1798  005620  052777  020000  174100          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1799  005626  012767  000012  173266          MOV     #10.,SHIFT      ;# OF SHIFTS
 1800  005634  012767  000252  173636          MOV     #252,$TMP1      ;DATA CHAR
 1801                                      ;NOTE:  THE ABOVE CHARACTER IS MISSING STOP BIT
 1802  005642  004767  011270                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1803  005646  016703  174044                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
 1804  005652  012700  120125                  MOV     #RXERR!FRMERR!125,R0    ;EXPECTED
 1805  005656  017701  174034                  MOV     @RXDBUF,R1      ;ACTUAL
 1806  005662  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
 1807  005664  001401                          BEQ     .+4
 1808  005666  104000                          ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
                                                          ;IF LOWER BYTE DOES NOT MATCH  IT
                                                          ;PROBABLY IS A LENGTH SELECT PROBLEM
 1809
 1810
 1811                                          ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
 1812                                          ;;OF THE RECEIVER
 1813                                          ;;MODE: ISOC (ISYMOD)
 1814                                          ;;PARITY: EVEPAR
 1815                                          ;;LENGTH: FIVE PLUS PARITY
 1816                                          ;;CHAR:  25
 1817                                          ;;
 1818                                      ;;*****************************************************************
 1819  005670  000004                      TST13:  SCOPE
 1820  005672  052777  000400  174026          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1821  005700  012777  000000  174014          MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1822  005706  052777  000400  174012          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1823
 1824                                      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1825  005714  012777  064001  174004          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1826
 1827                                      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1828  005722  012777  001400  173772          MOV     #ISYMOD!FIVE!EVEPAR!0,@PARCSR
 1829  005730  052777  000020  173754          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1830                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1831  005736  042777  020000  173762          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1832  005744  052777  020000  173754          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1833                                      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1834  005752  042777  020000  173746          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1835  005760  052777  020000  173740          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1836  005766  016703  173724                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1837  005772  012700  110025                  MOV     #RXERR!PARER!25,R0      ;EXPECTED
 1838  005776  012767  000010  173116          MOV     #8.,SHIFT       ;# OF SHIFTS
 1839  006004  012767  000252  173466          MOV     #252,$TMP1      ;DATA CHAR
 1840  006012  004767  011120                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1841  006016  105777  173670                  TSTB    @RXCSR  ;RXDONE ?
 1842  006022  100401                          BMI     .+4
 1843  006024  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
 1844  006026  017701  173664                  MOV     @RXDBUF,R1      ;ACTUAL
 1845  006032  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1846  006034  001401                          BEQ     .+4
```

```
1848  006036  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
1849                                                  ;NOTE THAT THE PARITY BIT SHOULD
1850                                                  ;SHOW UP IN THE DATA
1851                                                  ;IE. BIT FIVE FOR FIVE LEVEL CODE
1852
1853                                          ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1854                                          ;;OF THE RECEIVER
1855                                          ;;MODE: ISOC (ISYMOD)
1856                                          ;;PARITY: EVEPAR
1857                                          ;;LENGTH: SIX PLUS PARITY
1858                                          ;;CHAR:  25
1859                                          ;;
1860                                    ;;*************************************************************************
1861  006040  000004                    TST14:  SCOPE
1862  006042  052777  000400  173656            BIS     #MRESET,@TXCSR  ;MASTER RESET
1863  006050  012777  000000  173644            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1864  006056  052777  000400  173642            BIS     #MRESET,@TXCSR  ;MASTER RESET
1865
1866                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1867  006064  012777  064001  173634            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1868
1869                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1870  006072  012777  003400  173622            MOV     #ISYMOD!SIX!EVEPAR!0,@PARCSR
1871  006100  052777  000020  173604            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1872                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1873  006106  042777  020000  173612            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1874  006114  052777  020000  173604            BIS     #CLK,@TXCSR     ;POKE CLK UP
1875                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1876  006122  042777  020000  173576            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1877  006130  052777  020000  173570            BIS     #CLK,@TXCSR     ;POKE CLK UP
1878  006136  016703  173554                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1879  006142  012700  110025                    MOV     #RXERR!PARER!25,R0      ;EXPECTED
1880  006146  012767  000011  172746            MOV     #9.,SHIFT       ;# OF SHIFTS
1881  006154  012767  000452  173316            MOV     #452,$TMP1      ;DATA CHAR
1882  006162  004767  010750                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1883  006166  105777  173520                    TSTB    @RXCSR  ;RXDONE ?
1884  006172  100401                            BMI     .+4
1885  006174  104004                            ERROR   4               ;RXDONE SHOULD BE ASSERTED
1886  006176  017701  173514                    MOV     @RXDBUF,R1      ;ACTUAL
1887  006202  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1888  006204  001401                            BEQ     .+4
1889  006206  104000                            ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
1890                                                  ;NOTE THAT THE PARITY BIT SHOULD
1891                                                  ;SHOW UP IN THE DATA
1892                                                  ;IE. BIT SIX FOR SIX LEVEL CODE
1893
1894                                          ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1895                                          ;;OF THE RECEIVER
1896                                          ;;MODE: ISOC (ISYMOD)
1897                                          ;;PARITY: EVEPAR
1898                                          ;;LENGTH: SEVEN PLUS PARITY
1899                                          ;;CHAR:  325
1900                                          ;;
1901                                    ;;*************************************************************************
1902  006210  000004                    TST15:  SCOPE
1903  006212  052777  000400  173506            BIS     #MRESET,@TXCSR  ;MASTER RESET
```

```
1904  006220  012777  000000  173474          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1905  006226  052777  000400  173472          BIS     #MRESET,@TXCSR  ;MASTER RESET
1906
1907                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1908  006234  012777  064001  173464          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1909
1910                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1911  006242  012777  005400  173452          MOV     #ISYMOD!SEVEN!EVEPAR!0,@PARCSR
1912  006250  052777  000020  173434          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1913                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1914  006256  042777  020000  173442          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1915  006264  052777  020000  173434          BIS     #CLK,@TXCSR     ;POKE CLK UP
1916                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1917  006272  042777  020000  173426          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1918  006300  052777  020000  173420          BIS     #CLK,@TXCSR     ;POKE CLK UP
1919  006306  016703  173404                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1920  006312  012700  110325                  MOV     #RXERR!PARER!325,R0     ;EXPECTED
1921  006316  012767  000012  172576          MOV     #10.,SHIFT      ;# OF SHIFTS
1922  006324  012767  001652  173146          MOV     #1652,$TMP1     ;DATA CHAR
1923  006332  004767  010600                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1924  006336  105777  173350                  TSTB    @RXCSR  ;RXDONE ?
1925  006342  100401                          BMI     .+4
1926  006344  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
1927  006346  017701  173344                  MOV     @RXDBUF,R1      ;ACTUAL
1928  006352  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1929  006354  001401                          BEQ     .+4
1930  006356  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
1931                                                  ;NOTE THAT THE PARITY BIT SHOULD
1932                                                  ;SHOW UP IN THE DATA
1933                                                  ;IE.  BIT SEVEN FOR SEVEN LEVEL CODE
1934
1935                                          ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1936                                          ;;OF THE RECEIVER
1937                                          ;;MODE: ISOC (ISYMOD)
1938                                          ;;PARITY: EVEPAR
1939                                          ;;LENGTH: EIGHT PLUS PARITY
1940                                          ;;CHAR:  125
1941                                          ;;
1942                                  ;;*****************************************************************
1943  006360  000004                  TST16:  SCOPE
1944  006362  052777  000400  173336          BIS     #MRESET,@TXCSR  ;MASTER RESET
1945  006370  012777  000000  173324          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1946  006376  052777  000400  173322          BIS     #MRESET,@TXCSR  ;MASTER RESET
1947
1948                                  ;SET MAINT DATA,CLK,BREAK,&MA!NTENANCE MODE
1949  006404  012777  064001  173314          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1950
1951                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1952  006412  012777  007400  173302          MOV     #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
1953  006420  052777  000020  173264          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1954                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1955  006426  042777  020000  173272          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1956  006434  052777  020000  173264          BIS     #CLK,@TXCSR     ;POKE CLK UP
1957                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1958  006442  042777  020000  173256          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1959  006450  052777  020000  173250          BIS     #CLK,@TXCSR     ;POKE CLK UP
```

```
1960  006456  016703  173234          MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1961  006462  012700  110125          MOV     #RXERR!PARER!125,R0     ;EXPECTED
1962  006466  012767  000013  172426  MOV     #11.,SHIFT      ;# OF SHIFTS
1963  006474  012767  003252  172776  MOV     #3252,$TMP1     ;DATA CHAR
1964  006502  004767  010430          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1965  006506  105777  173200          TSTB    @RXCSR  ;RXDONE ?
1966  006512  100401                  BMI     .+4
1967  006514  104004                  ERROR   4               ;RXDONE SHOULD BE ASSERTED
1968  006516  017701  173174          MOV     @RXDBUF,R1      ;ACTUAL
1969  006522  020001                  CMP     R0,R1   ;COMPARE EXP VS. ACT
1970  006524  001401                  BEQ     .+4
1971  006526  104000                  ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
1972
1973                                  ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
1974                                  ;;OF THE RECEIVER
1975                                  ;;MODE: ISOC (ISYMOD)
1976                                  ;;PARITY: ODDPAR
1977                                  ;;LENGTH: FIVE PLUS PARITY
1978                                  ;;CHAR:  65
1979                                  ;;
1980                                ;;*****************************************************************
1981  006530  000004                TST17:  SCOPE
1982  006532  052777  000400  173166          BIS     #MRESET,@TXCSR  ;MASTER RESET
1983  006540  012777  000000  173154          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1984  006546  052777  000400  173152          BIS     #MRESET,@TXCSR  ;MASTER RESET
1985
1986                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1987  006554  012777  064001  173144          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1988
1989                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1990  006562  012777  001000  173132          MOV     #ISYMOD!FIVE!ODDPAR!0,@PARCSR
1991  006570  052777  000020  173114          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1992                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1993  006576  042777  020000  173122          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1994  006604  052777  020000  173114          BIS     #CLK,@TXCSR     ;POKE CLK UP
1995                              ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1996  006612  042777  020000  173106          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1997  006620  052777  020000  173100          BIS     #CLK,@TXCSR     ;POKE CLK UP
1998  006626  016703  173064          MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1999  006632  012700  110065          MOV     #RXERR!PARER!65,R0      ;EXPECTED
2000  006636  012767  000010  172256  MOV     #8.,SHIFT       ;# OF SHIFTS
2001  006644  012767  000352  172626  MOV     #352,$TMP1      ;DATA CHAR
2002  006652  004767  010260          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2003  006656  105777  173030          TSTB    @RXCSR  ;RXDONE ?
2004  006662  100401                  BMI     .+4
2005  006664  104004                  ERROR   4               ;RXDONE SHOULD BE ASSERTED
2006  006666  017701  173024          MOV     @RXDBUF,R1      ;ACTUAL
2007  006672  020001                  CMP     R0,R1   ;COMPARE EXP VS. ACT
2008  006674  001401                  BEQ     .+4
2009  006676  104000                  ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
2010                                                  ;NOTE THAT THE PARITY BIT SHOULD
2011                                                ;SHOW UP IN THE DATA
2012                                                ;IE. BIT FIVE FOR FIVE LEVEL CODE
2013
2014                                ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
2015                                ;;OF THE RECEIVER
```

```
2016                                            ;;MODE: ISOC (ISYMOD)
2017                                            ;;PARITY: ODDPAR
2018                                            ;;LENGTH: SIX PLUS PARITY
2019                                            ;;CHAR:  125
2020                                            ;;
2021                                    ;;**************************************************************
2022  006700  000004                   TST20:  SCOPE
2023  006702  052777  000400  173016            BIS     #MRESET,@TXCSR  ;MASTER RESET
2024  006710  012777  000000  173004            MOV     #ISYMOD,@PARCSR ;SET THE MODE
2025  006716  052777  000400  173002            BIS     #MRESET,@TXCSR  ;MASTER RESET
2026
2027                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2028  006724  012777  064001  172774            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2029
2030                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2031  006732  012777  003000  172762            MOV     #ISYMOD!SIX!ODDPAR!0,@PARCSR
2032  006740  052777  000020  172744            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2033                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2034  006746  042777  020000  172752            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2035  006754  052777  020000  172744            BIS     #CLK,@TXCSR     ;POKE CLK UP
2036                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2037  006762  042777  020000  172736            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2038  006770  052777  020000  172730            BIS     #CLK,@TXCSR     ;POKE CLK UP
2039  006776  016703  172714                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2040  007002  012700  110125                    MOV     #RXERR!PARER!125,R0     ;EXPECTED
2041  007006  012767  000011  172106            MOV     #9.,SHIFT       ;# OF SHIFTS
2042  007014  012767  000652  172456            MOV     #652,$TMP1      ;DATA CHAR
2043  007022  004767  010110                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2044  007026  105777  172660                    TSTB    @RXCSR  ;RXDONE ?
2045  007032  100401                            BMI     .+4
2046  007034  104004                            ERROR   4               ;RXDONE SHOULD BE ASSERTED
2047  007036  017701  172654                    MOV     @RXDBUF,R1      ;ACTUAL
2048  007042  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
2049  007044  001401                            BEQ     .+4
2050  007046  104000                            ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
2051                                                    ;NOTE THAT THE PARITY BIT SHOULD
2052                                                    ;SHOW UP IN THE DATA
2053                                                    ;IE. BIT SIX FOR SIX LEVEL CODE
2054
2055                                            ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
2056                                            ;;OF THE RECEIVER
2057                                            ;;MODE: ISOC (ISYMOD)
2058                                            ;;PARITY: ODDPAR
2059                                            ;;LENGTH: SEVEN PLUS PARITY
2060                                            ;;CHAR:  125
2061                                            ;;
2062                                    ;;**************************************************************
2063  007050  000004                   TST21:  SCOPE
2064  007052  052777  000400  172646            BIS     #MRESET,@TXCSR  ;MASTER RESET
2065  007060  012777  000000  172634            MOV     #ISYMOD,@PARCSR ;SET THE MODE
2066  007066  052777  000400  172632            BIS     #MRESET,@TXCSR  ;MASTER RESET
2067
2068                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2069  007074  012777  064001  172624            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2070
2071                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
```

```
2072  007102  012777  005000  172612          MOV     #ISYMOD!SEVEN!ODDPAR!0,@PARCSR
2073  007110  052777  000020  172574          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2074                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2075  007116  042777  020000  172602          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2076  007124  052777  020000  172574          BIS     #CLK,@TXCSR     ;POKE CLK UP
2077                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2078  007132  042777  020000  172566          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2079  007140  052777  020000  172560          BIS     #CLK,@TXCSR     ;POKE CLK UP
2080  007146  016703  172544                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2081  007152  012700  110125                  MOV     #RXERR!PARER!125,R0     ;EXPECTED
2082  007156  012767  000012  171736          MOV     #10.,SHIFT      ;# OF SHIFTS
2083  007164  012767  001252  172306          MOV     #1252,$TMP1     ;DATA CHAR
2084  007172  004767  007740                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2085  007176  105777  172510                  TSTB    @RXCSR  ;RXDONE ?
2086  007202  100401                          BMI     .+4
2087  007204  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
2088  007206  017701  172504                  MOV     @RXDBUF,R1      ;ACTUAL
2089  007212  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2090  007214  001401                          BEQ     .+4
2091  007216  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
2092                                                  ;NOTE THAT THE PARITY BIT SHOULD
2093                                                  ;SHOW UP IN THE DATA
2094                                                  ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
2095
2096                                          ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
2097                                          ;;OF THE RECEIVER
2098                                          ;;MODE: ISOC (ISYMOD)
2099                                          ;;PARITY: ODDPAR
2100                                          ;;LENGTH: EIGHT PLUS PARITY
2101                                          ;;CHAR:  125
2102                                          ;;
2103                                  ;;**************************************************************
2104  007220  000004                  TST22:  SCOPE
2105  007222  052777  000400  172476          BIS     #MRESET,@TXCSR  ;MASTER RESET
2106  007230  012777  000000  172464          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2107  007236  052777  000400  172462          BIS     #MRESET,@TXCSR  ;MASTER RESET
2108
2109                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2110  007244  012777  064001  172454          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2111
2112                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2113  007252  012777  007000  172442          MOV     #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
2114  007260  052777  000020  172424          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2115                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2116  007266  042777  020000  172432          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2117  007274  052777  020000  172424          BIS     #CLK,@TXCSR     ;POKE CLK UP
2118                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2119  007302  042777  020000  172416          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2120  007310  052777  020000  172410          BIS     #CLK,@TXCSR     ;POKE CLK UP
2121  007316  016703  172374                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2122  007322  012700  110125                  MOV     #RXERR!PARER!125,R0     ;EXPECTED
2123  007326  012767  000013  171566          MOV     #11.,SHIFT      ;# OF SHIFTS
2124  007334  012767  002252  172136          MOV     #2252,$TMP1     ;DATA CHAR
2125  007342  004767  007570                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2126  007346  105777  172340                  TSTB    @RXCSR  ;RXDONE ?
2127  007352  100401                          BMI     .+4
```

```
2128  007354  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
2129  007356  017701  172334                  MOV     @RXDBUF,R1      ;ACTUAL
2130  007362  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2131  007364  001401                          BEQ     .+4
2132  007366  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
2133
2134                                          ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
2135                                          ;;RECEIVER
2136                                          ;;LENGTH: EIGHT PLUS PARITY
2137                                          ;;MODE: ISYMOD
2138                                          ;;PARITY: EVEPAR
2139                                          ;;
2140                                  ;;**************************************************************
2141  007370  000004                  TST23:  SCOPE
2142  007372  052777  000400  172326          BIS     #MRESET,@TXCSR  ;MASTER RESET
2143  007400  012777  000000  172314          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2144  007406  052777  000400  172312          BIS     #MRESET,@TXCSR  ;MASTER RESET
2145
2146                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2147  007414  012777  064001  172304          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2148
2149                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2150  007422  012777  007400  172272          MOV     #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
2151  007430  052777  000020  172254          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2152                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2153  007436  042777  020000  172262          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2154  007444  052777  020000  172254          BIS     #CLK,@TXCSR     ;POKE CLK UP
2155                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2156  007452  042777  020000  172246          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2157  007460  052777  020000  172240          BIS     #CLK,@TXCSR     ;POKE CLK UP
2158  007466  016703  172224                  MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
2159  007472  005004                          CLR     R4      ;DATA CHAR
2160  007474  010400                  1$:     MOV     R4,R0   ;EXPECTED
2161  007476  012767  000013  171416          MOV     #11.,SHIFT      ;# OF SHIFTS
2162  007504  010467  171770                  MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
2163  007510  004767  007574                  JSR     PC,EVEN8        ;CALC PARITY
2164  007514  000241                          CLC
2165  007516  006167  171756                  ROL     $TMP1   ;GENERATE START BIT
2166  007522  052767  002000  171750          BIS     #BIT10,$TMP1    ;GENERATE STOP BIT
2167                                  ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2168  007530  004767  007402                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2169  007534  017701  172156                  MOV     @RXDBUF,R1      ;ACTUAL
2170  007540  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
2171  007542  001401                          BEQ     .+4
2172  007544  104002                          ERROR   2       ;DATA CHARS SHOULD MATCH
2173                                                          ;THERE SHOULD BE NO PARITY ERROR
2174  007546  005204                          INC     R4      ;UPGRADE NEXT CHAR
2175  007550  105704                          TSTB    R4      ;LAST CHAR ?
2176  007552  001350                          BNE     1$
2177
2178                                          ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
2179                                          ;;RECEIVER
2180                                          ;;LENGTH: EIGHT PLUS PARITY
2181                                          ;;MODE: ISYMOD
2182                                          ;;PARITY: ODDPAR
2183                                          ;;
```

```
 2184                                   ;;*****************************************************************
 2185  007554  000004                  TST24:  SCOPE
 2186  007556  052777  000400  172142          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2187  007564  012777  000000  172130          MOV     #ISYMOD,@PARCSR ;SET THE MODE
 2188  007572  052777  000400  172126          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2189
 2190                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 2191  007600  012777  064001  172120          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 2192
 2193                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 2194  007606  012777  007000  172106          MOV     #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
 2195  007614  052777  000020  172070          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 2196                                         ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 2197  007622  042777  020000  172076          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2198  007630  052777  020000  172070          BIS     #CLK,@TXCSR     ;POKE CLK UP
 2199                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 2200  007636  042777  020000  172062          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2201  007644  052777  020000  172054          BIS     #CLK,@TXCSR     ;POKE CLK UP
 2202  007652  016703  172040                  MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
 2203  007656  005004                          CLR     R4      ;DATA CHAR
 2204  007660  010400                  1$:     MOV     R4,R0   ;EXPECTED
 2205  007662  012767  000013  171232          MOV     #11.,SHIFT      ;# OF SHIFTS
 2206  007670  010467  171604                  MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
 2207  007674  004767  007324                  JSR     PC,ODD8 ;CALC PARITY
 2208  007700  000241                          CLC
 2209  007702  006167  171572                  ROL     $TMP1   ;GENERATE START BIT
 2210  007706  052767  002000  171564          BIS     #BIT10,$TMP1    ;GENERATE STOP BIT
 2211                                   ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
 2212  007714  004767  007216                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 2213  007720  017701  171772                  MOV     @RXDBUF,R1      ;ACTUAL
 2214  007724  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
 2215  007726  001401                          BEQ     .+4
 2216  007730  104002                          ERROR   2       ;DATA CHARS SHOULD MATCH
 2217                                                          ;THERE SHOULD BE NO PARITY ERROR
 2218  007732  005204                          INC     R4      ;UPGRADE NEXT CHAR
 2219  007734  105704                          TSTB    R4      ;LAST CHAR ?
 2220  007736  001350                          BNE     1$
 2221
 2222                                         ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
 2223                                         ;;RECEIVER
 2224                                         ;;LENGTH: EIGHT PLUS PARITY
 2225                                         ;;MODE: SYNEXT
 2226                                         ;;PARITY: EVEPAR
 2227                                         ;;
 2228                                   ;;*****************************************************************
 2229  007740  000004                  TST25:  SCOPE
 2230  007742  052777  000400  171756          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2231  007750  012777  020000  171744          MOV     #SYNEXT,@PARCSR ;SET THE MODE
 2232  007756  052777  000400  171742          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2233
 2234                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 2235  007764  012777  064001  171734          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 2236
 2237                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 2238  007772  012777  027400  171722          MOV     #SYNEXT!EIGHT!EVEPAR!0,@PARCSR
 2239  010000  052777  000020  171704          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
```

```
2240                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2241  010006  042777  020000  171712            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2242  010014  052777  020000  171704            BIS     #CLK,@TXCSR     ;POKE CLK UP
2243  010022  016703  171670                    MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
2244  010026  005004                            CLR     R4      ;DATA CHAR
2245  010030  010400                    1$:     MOV     R4,R0   ;EXPECTED
2246  010032  012767  000011  171062            MOV     #9.,SHIFT       ;# OF SHIFTS
2247  010040  010467  171434                    MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
2248  010044  004767  007240                    JSR     PC,EVEN8        ;CALC PARITY
2249                                    ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2250  010050  004767  007062                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2251  010054  017701  171636                    MOV     @RXDBUF,R1      ;ACTUAL
2252  010060  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
2253  010062  001401                            BEQ     .+4
2254  010064  104002                            ERROR   2       ;DATA CHARS SHOULD MATCH
2255                                                            ;THERE SHOULD BE NO PARITY ERROR
2256  010066  005204                            INC     R4      ;UPGRADE NEXT CHAR
2257  010070  105704                            TSTB    R4      ;LAST CHAR ?
2258  010072  001356                            BNE     1$
2259
2260                                            ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
2261                                            ;;RECEIVER
2262                                            ;;LENGTH: EIGHT PLUS PARITY
2263                                            ;;MODE: SYNEXT
2264                                            ;;PARITY: ODDPAR
2265                                            ;;
2266                                    ;;**********************************************************************
2267  010074  000004                    TST26:  SCOPE
2268  010076  052777  000400  171622            BIS     #MRESET,@TXCSR  ;MASTER RESET
2269  010104  012777  020000  171610            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2270  010112  052777  000400  171606            BIS     #MRESET,@TXCSR  ;MASTER RESET
2271
2272                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2273  010120  012777  064001  171600            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2274
2275                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2276  010126  012777  027000  171566            MOV     #SYNEXT!EIGHT!ODDPAR!0,@PARCSR
2277  010134  052777  000020  171550            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2278                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2279  010142  042777  020000  171556            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2280  010150  052777  020000  171550            BIS     #CLK,@TXCSR     ;POKE CLK UP
2281  010156  016703  171534                    MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
2282  010162  005004                            CLR     R4      ;DATA CHAR
2283  010164  010400                    1$:     MOV     R4,R0   ;EXPECTED
2284  010166  012767  000011  170726            MOV     #9.,SHIFT       ;# OF SHIFTS
2285  010174  010467  171300                    MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
2286  010200  004767  007020                    JSR     PC,ODD8 ;CALC PARITY
2287                                    ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2288  010204  004767  006726                    JSR     PC,RPOKE        ;SHIFT !N THIS CHAR
2289  010210  017701  171502                    MOV     @RXDBUF,R1      ;ACTUAL
2290  010214  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
2291  010216  001401                            BEQ     .+4
2292  010220  104002                            ERROR   2       ;DATA CHARS SHOULD MATCH
2293                                                            ;THERE SHOULD BE NO PARITY ERROR
2294  010222  005204                            INC     R4      ;UPGRADE NEXT CHAR
2295  010224  105704                            TSTB    R4      ;LAST CHAR ?
```

```
  2296  010226  001356                          BNE     1$
  2297
  2298                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
  2299                                          ;;OF THE RECEIVER LOGIC
  2300                                          ;;MODE: ISYMOD
  2301                                          ;;LENGTH: FIVE
  2302                                          ;;NOTE:  RXDONE SHOULD NEVER ASSERT
  2303                                          ;;CHAR:  26 (SYNC)
  2304                                          ;;
  2305                                  ;;**************************************************************************
  2306  010230  000004                 TST27:   SCOPE
  2307  010232  052777  000400  171466          BIS     #MRESET,@TXCSR  ;MASTER RESET
  2308  010240  012777  000000  171454          MOV     #ISYMOD,@PARCSR ;SET THE MODE
  2309  010246  052777  000400  171452          BIS     #MRESET,@TXCSR  ;MASTER RESET
  2310
  2311                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
  2312  010254  012777  064001  171444          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
  2313
  2314                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
  2315  010262  012777  000026  171432          MOV     #ISYMOD!FIVE!NOPAR!26,@PARCSR
  2316  010270  052777  000020  171414          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
  2317                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
  2318  010276  042777  020000  171422          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
  2319  010304  052777  020000  171414          BIS     #CLK,@TXCSR     ;POKE CLK UP
  2320                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
  2321  010312  042777  020000  171406          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
  2322  010320  052777  020000  171400          BIS     #CLK,@TXCSR     ;POKE CLK UP
  2323  010326  052777  000400  171356          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
  2324  010334  012767  000003  170562          MOV     #3,COUNT        ;# OF SYNC CHARS
  2325  010342  012767  000154  171130  1$:     MOV     #154,$TMP1      ;CHAR TO BE SHIFTED
  2326  010350  012767  000007  170544          MOV     #7,SHIFT        ;# OF SHIFTS
  2327  010356  004767  006554                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
  2328  010362  105777  171324                  TSTB    @RXCSR  ;RXDONE ?
  2329  010366  100001                          BPL     .+4
  2330  010370  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
  2331  010372  005367  170526                  DEC     COUNT   ;# OF SYNC CHARS
  2332  010376  001361                          BNE     1$
  2333
  2334                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
  2335                                          ;;OF THE RECEIVER LOGIC
  2336                                          ;;MODE: ISYMOD
  2337                                          ;;LENGTH: SIX
  2338                                          ;;NOTE:  RXDONE SHOULD NEVER ASSERT
  2339                                          ;;CHAR:  26 (SYNC)
  2340                                          ;;
  2341                                  ;;**************************************************************************
  2342  010400  000004                 TST30:   SCOPE
  2343  010402  052777  000400  171316          BIS     #MRESET,@TXCSR  ;MASTER RESET
  2344  010410  012777  000000  171304          MOV     #ISYMOD,@PARCSR ;SET THE MODE
  2345  010416  052777  000400  171302          BIS     #MRESET,@TXCSR  ;MASTER RESET
  2346
  2347                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
  2348  010424  012777  064001  171274          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
  2349
  2350                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
  2351  010432  012777  002026  171262          MOV     #ISYMOD!SIX!NOPAR!26,@PARCSR
```

```
2352  010440  052777  000020  171244          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2353                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2354  010446  042777  020000  171252          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2355  010454  052777  020000  171244          BIS     #CLK,@TXCSR     ;POKE CLK UP
2356                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2357  010462  042777  020000  171236          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2358  010470  052777  020000  171230          BIS     #CLK,@TXCSR     ;POKE CLK UP
2359  010476  052777  000400  171206          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2360  010504  012767  000003  170412          MOV     #3,COUNT        ;# OF SYNC CHARS
2361  010512  012767  000254  170760  1$:     MOV     #254,$TMP1      ;CHAR TO BE SHIFTED
2362  010520  012767  000010  170374          MOV     #8.,SHIFT       ;# OF SHIFTS
2363  010526  004767  006404                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2364  010532  105777  171154                  TSTB    @RXCSR  ;RXDONE ?
2365  010536  100001                          BPL     .+4
2366  010540  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2367  010542  005367  170356                  DEC     COUNT   ;# OF SYNC CHARS
2368  010546  001361                          BNE     1$
2369
2370                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2371                                          ;;OF THE RECEIVER LOGIC
2372                                          ;;MODE: ISYMOD
2373                                          ;;LENGTH: SEVEN
2374                                          ;;NOTE:  RXDONE SHOULD NEVER ASSERT
2375                                          ;;CHAR:  26 (SYNC)
2376                                          ;;
2377                                  ;;*****************************************************************
2378  010550  000004                  TST31:  SCOPE
2379  010552  052777  000400  171146          BIS     #MRESET,@TXCSR  ;MASTER RESET
2380  010560  012777  000000  171134          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2381  010566  052777  000400  171132          BIS     #MRESET,@TXCSR  ;MASTER RESET
2382
2383                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2384  010574  012777  064001  171124          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2385
2386                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2387  010602  012777  004026  171112          MOV     #ISYMOD!SEVEN!NOPAR!26,@PARCSR
2388  010610  052777  000020  171074          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2389                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2390  010616  042777  020000  171102          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2391  010624  052777  020000  171074          BIS     #CLK,@TXCSR     ;POKE CLK UP
2392                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2393  010632  042777  020000  171066          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2394  010640  052777  020000  171060          BIS     #CLK,@TXCSR     ;POKE CLK UP
2395  010646  052777  000400  171036          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2396  010654  012767  000003  170242          MOV     #3,COUNT        ;# OF SYNC CHARS
2397  010662  012767  000454  170610  1$:     MOV     #454,$TMP1      ;CHAR TO BE SHIFTED
2398  010670  012767  000011  170224          MOV     #9.,SHIFT       ;# OF SHIFTS
2399  010676  004767  006234                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2400  010702  105777  171004                  TSTB    @RXCSR  ;RXDONE ?
2401  010706  100001                          BPL     .+4
2402  010710  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2403  010712  005367  170206                  DEC     COUNT   ;# OF SYNC CHARS
2404  010716  001361                          BNE     1$
2405
2406                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2407                                          ;;OF THE RECEIVER LOGIC
```

```
                                        ;;MODE: ISYMOD
                                        ;;LENGTH: EIGHT
                                        ;;NOTE:  RXDONE SHOULD NEVER ASSERT
                                        ;;CHAR:  26 (SYNC)
                                        ;;
                                ;;*****************************************************************
010720  000004                 TST32:   SCOPE
010722  052777  000400  170776          BIS     #MRESET,@TXCSR  ;MASTER RESET
010730  012777  000000  170764          MOV     #ISYMOD,@PARCSR ;SET THE MODE
010736  052777  000400  170762          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
010744  012777  064001  170754          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
010752  012777  006026  170742          MOV     #ISYMOD!EIGHT!NOPAR!26,@PARCSR
010760  052777  000020  170724          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                        ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
010766  042777  020000  170732          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010774  052777  020000  170724          BIS     #CLK,@TXCSR     ;POKE CLK UP

                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
011002  042777  020000  170716          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
011010  052777  020000  170710          BIS     #CLK,@TXCSR     ;POKE CLK UP
011016  052777  000400  170666          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
011024  012767  000003  170072          MOV     #3,COUNT        ;# OF SYNC CHARS
011032  012767  001054  170440  1$:     MOV     #1054,$TMP1     ;CHAR TO BE SHIFTED
011040  012767  000012  170054          MOV     #10.,SHIFT      ;# OF SHIFTS
011046  004767  006064                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
011052  105777  170634                  TSTB    @RXCSR  ;RXDONE ?
011056  100001                          BPL     .+4
011060  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
011062  005367  170036                  DEC     COUNT   ;# OF SYNC CHARS
011066  001361                          BNE     1$

                                        ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
                                        ;;OF THE RECEIVER LOGIC
                                        ;;MODE: SYNEXT
                                        ;;LENGTH: FIVE
                                        ;;NOTE:  RXDONE SHOULD NEVER ASSERT
                                        ;;CHAR:  26 (SYNC)
                                        ;;
                                ;;*****************************************************************
011070  000004                 TST33:   SCOPE
011072  052777  000400  170626          BIS     #MRESET,@TXCSR  ;MASTER RESET
011100  012777  020000  170614          MOV     #SYNEXT,@PARCSR ;SET THE MODE
011106  052777  000400  170612          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
011114  012777  064001  170604          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
011122  012777  020026  170572          MOV     #SYNEXT!FIVE!NOPAR!26,@PARCSR
011130  052777  000020  170554          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
011136  042777  020000  170562          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
011144  052777  020000  170554          BIS     #CLK,@TXCSR     ;POKE CLK UP
```

```
2464  011152  052777  000400  170532          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2465  011160  012767  000003  167736          MOV     #3,COUNT        ;# OF SYNC CHARS
2466  011166  012767  000026  170304  1$:     MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2467  011174  012767  000005  167720          MOV     #5,SHIFT        ;# OF SHIFTS
2468  011202  004767  005730                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2469  011206  105777  170500                  TSTB    @RXCSR  ;RXDONE ?
2470  011212  100001                          BPL     .+4
2471  011214  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2472  011216  005367  167702                  DEC     COUNT   ;# OF SYNC CHARS
2473  011222  001361                          BNE     1$
2474
2475                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2476                                          ;;OF THE RECEIVER LOGIC
2477                                          ;;MODE: SYNEXT
2478                                          ;;LENGTH: SIX
2479                                          ;;NOTE:  RXDONE SHOULD NEVER ASSERT
2480                                          ;;CHAR:  26 (SYNC)
2481                                          ;;
2482                                  ;;**************************************************************
2483  011224  000004                  TST34:  SCOPE
2484  011226  052777  000400  170472          BIS     #MRESET,@TXCSR  ;MASTER RESET
2485  011234  012777  020000  170460          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2486  011242  052777  000400  170456          BIS     #MRESET,@TXCSR  ;MASTER RESET
2487
2488                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2489  011250  012777  064001  170450          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2490
2491                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2492  011256  012777  022026  170436          MOV     #SYNEXT!SIX!NOPAR!26,@PARCSR
2493  011264  052777  000020  170420          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2494                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2495  011272  042777  020000  170426          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2496  011300  052777  020000  170420          BIS     #CLK,@TXCSR     ;POKE CLK UP
2497  011306  052777  000400  170376          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2498  011314  012767  000003  167602          MOV     #3,COUNT        ;# OF SYNC CHARS
2499  011322  012767  000026  170150  1$:     MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2500  011330  012767  000006  167564          MOV     #6,SHIFT        ;# OF SHIFTS
2501  011336  004767  005574                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2502  011342  105777  170344                  TSTB    @RXCSR  ;RXDONE ?
2503  011346  100001                          BPL     .+4
2504  011350  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2505  011352  005367  167546                  DEC     COUNT   ;# OF SYNC CHARS
2506  011356  001361                          BNE     1$
2507
2508                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2509                                          ;;OF THE RECEIVER LOGIC
2510                                          ;;MODE: SYNEXT
2511                                          ;;LENGTH: SEVEN
2512                                          ;;NOTE:  RXDONE SHOULD NEVER ASSERT
2513                                          ;;CHAR:  26 (SYNC)
2514                                          ;;
2515                                  ;;**************************************************************
2516  011360  000004                  TST35:  SCOPE
2517  011362  052777  000400  170336          BIS     #MRESET,@TXCSR  ;MASTER RESET
2518  011370  012777  020000  170324          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2519  011376  052777  000400  170322          BIS     #MRESET,@TXCSR  ;MASTER RESET
```

```
2520
2521                                            ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2522  011404  012777  064001  170314                    MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2523
2524                                            ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2525  011412  012777  024026  170302                    MOV     #SYNEXT!SEVEN!NOPAR!26,@PARCSR
2526  011420  052777  000020  170264                    BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2527                                            ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2528  011426  042777  020000  170272                    BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2529  011434  052777  020000  170264                    BIS     #CLK,@TXCSR     ;POKE CLK UP
2530  011442  052777  000400  170242                    BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2531  011450  012767  000003  167446                    MOV     #3,COUNT        ;# OF SYNC CHARS
2532  011456  012767  000026  170014  1$:               MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2533  011464  012767  000007  167430                    MOV     #7,SHIFT        ;# OF SHIFTS
2534  011472  004767  005440                          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2535  011476  105777  170210                          TSTB    @RXCSR  ;RXDONE ?
2536  011502  100001                                  BPL     .+4
2537  011504  104004                                  ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2538  011506  005367  167412                          DEC     COUNT   ;# OF SYNC CHARS
2539  011512  001361                                  BNE     1$
2540
2541                                                    ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2542                                                    ;;OF THE RECEIVER LOGIC
2543                                                    ;;MODE: SYNEXT
2544                                                    ;;LENGTH: EIGHT
2545                                                    ;;NOTE:  RXDONE SHOULD NEVER ASSERT
2546                                                    ;;CHAR:  26 (SYNC)
2547                                                    ;;
2548                                            ;;**********************************************************************
2549  011514  000004                          TST36:  SCOPE
2550  011516  052777  000400  170202                    BIS     #MRESET,@TXCSR  ;MASTER RESET
2551  011524  012777  020000  170170                    MOV     #SYNEXT,@PARCSR ;SET THE MODE
2552  011532  052777  000400  170166                    BIS     #MRESET,@TXCSR  ;MASTER RESET
2553
2554                                            ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2555  011540  012777  064001  170160                    MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2556
2557                                            ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2558  011546  012777  026026  170146                    MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2559  011554  052777  000020  170130                    BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2560                                            ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2561  011562  042777  020000  170136                    BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2562  011570  052777  020000  170130                    BIS     #CLK,@TXCSR     ;POKE CLK UP
2563  011576  052777  000400  170106                    BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2564  011604  012767  000003  167312                    MOV     #3,COUNT        ;# OF SYNC CHARS
2565  011612  012767  000026  167660  1$:               MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2566  011620  012767  000010  167274                    MOV     #8.,SHIFT       ;# OF SHIFTS
2567  011626  004767  005304                          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2568  011632  105777  170054                          TSTB    @RXCSR  ;RXDONE ?
2569  011636  100001                                  BPL     .+4
2570  011640  104004                                  ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2571  011642  005367  167256                          DEC     COUNT   ;# OF SYNC CHARS
2572  011646  001361                                  BNE     1$
2573
2574                                                    ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2575                                                    ;;OF THE RECEIVER LOGIC
```

```
2576                                            ;;MODE: SYNINT
2577                                            ;;LENGTH: FIVE
2578                                            ;;NOTE: RXDONE SHOULD NEVER ASSERT
2579                                            ;;CHAR: 26 (SYNC)
2580                                            ;;
2581                                    ;;**********************************************************************
2582  011650  000004                    TST37:  SCOPE
2583  011652  052777  000400  170046            BIS     #MRESET,@TXCSR  ;MASTER RESET
2584  011660  012777  030000  170034            MOV     #SYNINT,@PARCSR ;SET THE MODE
2585  011666  052777  000400  170032            BIS     #MRESET,@TXCSR  ;MASTER RESET
2586
2587                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2588  011674  012777  064001  170024            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2589
2590                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2591  011702  012777  030026  170012            MOV     #SYNINT!FIVE!NOPAR!26,@PARCSR
2592  011710  052777  000020  167774            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2593                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION...
2594  011716  042777  020000  170002            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2595  011724  052777  020000  167774            BIS     #CLK,@TXCSR     ;POKE CLK UP
2596                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2597  011732  042777  020000  167766            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2598  011740  052777  020000  167760            BIS     #CLK,@TXCSR     ;POKE CLK UP
2599  011746  052777  000400  167736            BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2600  011754  012767  000003  167142            MOV     #3,COUNT        ;# OF SYNC CHARS
2601  011762  012767  000026  167510  1$:       MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2602  011770  012767  000005  167124            MOV     #5,SHIFT        ;# OF SHIFTS
2603  011776  004767  005134                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2604  012002  105777  167704                    TSTB    @RXCSR  ;RXDONE ?
2605  012006  100001                            BPL     .+4
2606  012010  104004                            ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2607  012012  005367  167106                    DEC     COUNT   ;# OF SYNC CHARS
2608  012016  001361                            BNE     1$
2609
2610                                            ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2611                                            ;;OF THE RECEIVER LOGIC
2612                                            ;;MODE: SYNINT
2613                                            ;;LENGTH: SIX
2614                                            ;;NOTE: RXDONE SHOULD NEVER ASSERT
2615                                            ;;CHAR: 26 (SYNC)
2616                                            ;;
2617                                    ;;**********************************************************************
2618  012020  000004                    TST40:  SCOPE
2619  012022  052777  000400  167676            BIS     #MRESET,@TXCSR  ;MASTER RESET
2620  012030  012777  030000  167664            MOV     #SYNINT,@PARCSR ;SET THE MODE
2621  012036  052777  000400  167662            BIS     #MRESET,@TXCSR  ;MASTER RESET
2622
2623                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2624  012044  012777  064001  167654            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2625
2626                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2627  012052  012777  032026  167642            MOV     #SYNINT!SIX!NOPAR!26,@PARCSR
2628  012060  052777  000020  167624            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2629                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION...
2630  012066  042777  020000  167632            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2631  012074  052777  020000  167624            BIS     #CLK,@TXCSR     ;POKE CLK UP
```

```
2632                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2633  012102  042777  020000  167616            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2634  012110  052777  020000  167610            BIS     #CLK,@TXCSR     ;POKE CLK UP
2635  012116  052777  000400  167566            BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2636  012124  012767  000003  166772            MOV     #3,COUNT        ;# OF SYNC CHARS
2637  012132  012767  000026  167340  1$:       MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2638  012140  012767  000006  166754            MOV     #6,SHIFT        ;# OF SHIFTS
2639  012146  004767  004764                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2640  012152  105777  167534                    TSTB    @RXCSR  ;RXDONE ?
2641  012156  100001                            BPL     .+4
2642  012160  104004                            ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2643  012162  005367  166736                    DEC     COUNT   ;# OF SYNC CHARS
2644  012166  001361                            BNE     1$
2645
2646                                            ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2647                                            ;;OF THE RECEIVER LOGIC
2648                                            ;;MODE: SYNINT
2649                                            ;;LENGTH: SEVEN
2650                                            ;;NOTE:  RXDONE SHOULD NEVER ASSERT
2651                                            ;;CHAR:  26 (SYNC)
2652                                            ;;
2653                                    ;;**********************************************************************
2654  012170  000004                    TST41:  SCOPE
2655  012172  052777  000400  167526            BIS     #MRESET,@TXCSR  ;MASTER RESET
2656  012200  012777  030000  167514            MOV     #SYNINT,@PARCSR ;SET THE MODE
2657  012206  052777  000400  167512            BIS     #MRESET,@TXCSR  ;MASTER RESET
2658
2659                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2660  012214  012777  064001  167504            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2661
2662                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2663  012222  012777  034026  167472            MOV     #SYNINT!SEVEN!NOPAR!26,@PARCSR
2664  012230  052777  000020  167454            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2665                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2666  012236  042777  020000  167462            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2667  012244  052777  020000  167454            BIS     #CLK,@TXCSR     ;POKE CLK UP
2668                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2669  012252  042777  020000  167446            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2670  012260  052777  020000  167440            BIS     #CLK,@TXCSR     ;POKE CLK UP
2671  012266  052777  000400  167416            BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2672  012274  012767  000003  166622            MOV     #3,COUNT        ;# OF SYNC CHARS
2673  012302  012767  000026  167170  1$:       MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2674  012310  012767  000007  166604            MOV     #7,SHIFT        ;# OF SHIFTS
2675  012316  004767  004614                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2676  012322  105777  167364                    TSTB    @RXCSR  ;RXDONE ?
2677  012326  100001                            BPL     .+4
2678  012330  104004                            ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2679  012332  005367  166566                    DEC     COUNT   ;# OF SYNC CHARS
2680  012336  001361                            BNE     1$
2681
2682
2683                                            ;END OF PASS
2684                                            ;TYPE NAME OF TEST
2685                                            ;UPDATE PASS COUNT
2686                                            ;CHECK FOR EXIT TO ACT-11
2687                                            ;RESTART TEST
```

```
2688
2689  012340  000004                    .EOP:   SCOPE
2690  012342  004767  000340                    JSR     PC,CKSWR
2691  012346  104401                            TYPE                            ;TYPE NAME OF TEST
2692  012350  015476                            MEPASS
2693  012352  104413  012604                    CONVRT  ,OUTCRY
2694  012356  104401  015315                    TYPE    ,DEVICE
2695  012362  105767  166564                    TSTB    MULTD   ;ARE YOU RUNNING MULTIPLE DEVICES ?
2696  012366  001511                            BEQ     CCC     ;NO, JUMP AROUND
2697  012370  005767  166572                    TST     ACTREG  ;ARE ANY DEVICES ACTIVE ?
2698  012374  001007                            BNE     RUNIT   ;YES
2699  012376  104401  015327                    TYPE    ,MCOW   ;NO
2700  012402  016700  166560                    MOV     ACTREG,R0       ;DISPLAY ACTREG
2701  012406  000000                            HALT            ;SELECT SOMETHING TO RUN @ ACTREG:
2702                                            ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2703  012410  000167  167536                    JMP     .START  ;START OVER AGAIN..... YOU DESELECTED EVERYTHING
2704  012414  062767  000010  166532  RUNIT:   ADD     #10,BASEADD     ;NEXT BLOCK (ADDRESSES)
2705  012422  062767  000010  166532  ZERO:    ADD     #10,BASEIV      ;NEXT BLOCK (VECTORS)
2706  012430  000241                            CLC
2707  012432  006167  166532                    ROL     ROTADD  ;UP DATE ROTATING POINTER
2708  012436  103410                            BCS     2$      ;IS IT THE LAST DEVICE
2709                                            ;TO BE TESTED IN THIS PASS ?
2710  012440  036767  166524  166520            BIT     ROTADD,ACTREG   ;TEST THIS DEVICE FOR ACTIVE STATUS
2711  012446  001762                            BEQ     RUNIT   ;IF NOT ACTIVE, TRY NEXT ADDRESS
2712  012450  004767  000034                    JSR     PC,REPLAY       ;CALCULATE NEW PARAMETERS
2713  012454  000167  000210                    JMP     RESTRT  ;YES IT WAS ACTIVE,TEST THIS DEVICE
2714  012460  012767  000001  166502  2$:      MOV     #1,ROTADD       ;OK!,NOW SET UP  ROTATING
2715                                            ;POINTER FOR NEXT MULTIPLE PASS
2716  012466  016767  166464  166460            MOV     KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2717  012474  016767  166464  166460            MOV     KEEPIV,BASEIV   ;RESTORE BASE INTERRUPT VECTORS
2718  012502  004767  000002                    JSR     PC,REPLAY       ;CALC NEW PARAMETERS
2719  012506  000441                            BR      CCC     ;JUMP AROUND REPLAY
2720  012510  016767  166440  004416  REPLAY:  MOV     BASEADD,DUBASE  ;SET UP FOR NEW ADDRESSES
2721  012516  004767  004260                    JSR     PC,DUADDR       ;CREATE NEW ADDRESSES
2722  012522  016767  166434  167206            MOV     BASEIV,DURIV    ;CREATE DURIV
2723  012530  062767  000002  166424            ADD     #2,BASEIV
2724  012536  016767  166420  167174            MOV     BASEIV,DURIS    ;CREATE DURIS
2725  012544  062767  000002  166410            ADD     #2,BASEIV
2726  012552  016767  166404  167162            MOV     BASEIV,DUTIV    ;CREATE DUTIV
2727  012560  062767  000002  166374            ADD     #2,BASEIV
2728  012566  016767  166370  167150            MOV     BASEIV,DUTIS    ;CREATE DUTIS
2729  012574  016767  167136  166360            MOV     DURIV,BASEIV    ;RESTORE
2730  012602  000207                            RTS     PC
2731
2732  012604  000001                    OUTCRY: 1
2733  012606     006     002                    .BYTE   6,2
2734  012610  001712                            RXCSR
2735
2736  012612                            CCC:
2737  012612  005067  166564                    CLR     $TSTNM                  ,CLEAR TEST NUMBER
2738  012616  005067  166574                    CLR     $ERRPC                  ;CLEAR LAST ERROR PC
2739  012622  005067  166555                    CLR     $ERFLG                  ;CLEAR ERROR FLAG
2740  012626  005267  166260                    INC     PASCNT                  ;UPDATE PASS COUNT
2741  012632  016767  166254  166242            MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
2742  012640  016767  166246  166666            MOV     PASCNT,$PASS            ;PASS COUNT TO APT
2743  012646  013701  000042                    MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
```

```
2744  012652  001406                          BEQ     RESTRT          ;IF NO CONTINUE TESTING
2745  012654  000005                          RESET
2746  012656  000005                          RESET
2747  012660  004711                  $ENDAD: JSR     PC,(R1)
2748  012662  000240                          NOP
2749  012664  000240                          NOP
2750  012666  000240                          NOP
2751  012670                           RESTRT:
2752  012670  012767  003376  166510          MOV     #TST1+2,$LPADR  ;LOAD LAST ADDR
2753  012676  004767  000004                  JSR     PC,CKSWR
2754  012702  000167  170402                  JMP     .BEGIN
2755
2756                                          ;CHECK SWITCH REGISTER ROUTINE.
2757                                          ;CHECKS TO ALLOW FOR <G> TO ALLOW
2758                                          ;THE CHANGING OF LOCATION 176
2759
2760  012706  005737  000042          CKSWR:  TST     @#42
2761  012712  001040                          BNE     OUT
2762  012714  022767  000176  166516          CMP     #SWREG,SWR      ;SOFTWARE SWR PRESENT?
2763  012722  001034                          BNE     OUT             ;NO--LEAVE
2764  012724  105777  166514                  TSTB    @$TKS           ;CHECK TTY READY
2765  012730  100031                          BPL     OUT             ;NO--LEAVE
2766  012732  017767  166510  000422          MOV     @$TKB,..MSG     ;GET CHARACTER
2767  012740  042767  177600  000414          BIC     #177600,..MSG   ;STRIP JUNK
2768  012746  122767  000007  000406          CMPB    #7,..MSG        ;IS IT <G> ?
2769  012754  001017                          BNE     OUT             ;NO
2770  012756  104401  016103                  TYPE    ,MCNTG
2771  012762  005137  013022          CNTLU:  COM     @#RDSW
2772  012766  104401  016113                  TYPE    ,MMSWR
2773  012772  104413                          CONVRT
2774  012774  013024                          SWREGL
2775  012776  104406  016124                  INSTR,MMNEW
2776  013002  104410                          PARAM
2777  013004  000000                          0
2778  013006  177777                          177777
2779  013010  000176                          SWREG
2780  013012     000      001          .BYTE   0,1
2781  013014  005037  013022          OUT:    CLR     @#RDSW
2782  013020  000207                          RTS     PC
2783  013022  000000                  RDSW:   .WORD   0
2784  013024  000001                  SWREGL: 1
2785  013026     006      002                  .BYTE   6,2
2786  013030  000176                          SWREG
2787
2788  013032  000005                          5
2789
2790                                          ;CHECK FOR FREEZE ON CURRENT DATA
2791
2792  013034  004767  177646          .SCOP1: JSR     PC,CKSWR
2793  013040  032777  001000  166372          BIT     #SW09,@SWR
2794  013046  001402                          BEQ     1$
2795  013050  016716  166034                  MOV     LOCK,(SP)
2796  013054  000002                  1$:     RTI
2797                                  .SBTTL  TYPE ROUTINE
2798
2799                                  ;;**********************************************************************
```

```
 2800                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 2801                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 2802                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 2803                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 2804                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 2805                                   ;*
 2806                                   ;*CALL:
 2807                                   ;*1) USING A TRAP INSTRUCTION
 2808                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 2809                                   ;*OR
 2810                                   ;*      TYPE
 2811                                   ;*      MESADR
 2812                                   ;*
 2813
 2814  013056  105767  166375           $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 2815  013062  100002                           BPL     1$              ;;BR IF YES
 2816  013064  000000                           HALT                    ;;HALT HERE IF NO TERMINAL
 2817  013066  000430                           BR      3$              ;;LEAVE
 2818  013070  010046                   1$:     MOV     R0,-(SP)        ;;SAVE R0
 2819  013072  017600  000002                   MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 2820  013076  122767  000001  166442           CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 2821  013104  001011                           BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 2822  013106  132767  000100  166433           BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 2823  013114  001405                           BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 2824  013116  010067  000004                   MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 2825  013122  004767  000006'                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 2826  013126  000000                   61$:    .WORD   0               ;;MESSAGE ADDRESS
 2827  013130  132767  000040  166411   62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 2828  013136  001003                           BNE     60$             ;;YES,SKIP TYPE OUT
 2829  013140  112046                   2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 2830  013142  001005                           BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 2831  013144  005726                           TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 2832  013146  012600                   60$:    MOV     (SP)+,R0        ;;RESTORE R0
 2833  013150  062716  000002           3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 2834  013154  000002                           RTI                     ;;RETURN
 2835  013156  122716  000011           4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 2836  013162  001430                           BEQ     8$
 2837  013164  122716  000200                   CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 2838  013170  001006                           BNE     5$
 2839  013172  005726                           TST     (SP)+           ;;POP  <CR><LF> EQUIV
 2840  013174  104401                           TYPE                    ;;TYPE A CR AND LF
 2841  013176  001523                           $CRLF
 2842  013200  105067  000130                   CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 2843  013204  000755                           BR      2$              ;;GET NEXT CHARACTER
 2844  013206  004767  000056           5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 2845  013212  126726  166240           6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 2846  013216  001350                           BNE     2$              ;;IF NO GO GET NEXT CHAR.
 2847  013220  016746  166230                   MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 2848                                           ;;AND THE NULL CHAR.
 2849  013224  105366  000001           7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 2850  013230  002770                           BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 2851  013232  004767  000032                   JSR     PC,$TYPEC       ;;GO TYPE A NULL
 2852  013236  105367  000072                   DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 2853  013242  000770                           BR      7$              ;;LOOP
 2854
 2855                                   ;HORIZONTAL TAB PROCESSOR
```

```
2856
2857  013244  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2858  013250  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2859  013254  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2860  013262  001372                          BNE     9$              ;;TAB STOP
2861  013264  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
2862  013266  000724                          BR      2$              ;;GET NEXT CHARACTER
2863  013270  105777  166154          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2864  013274  100375                          BPL     $TYPEC
2865  013276  116677  000002  166146          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG
2866  013304  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2867  013312  001003                          BNE     1$              ;;BRANCH IF NO
2868  013314  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2869  013320  000406                          BR      $TYPEX          ;;EXIT
2870  013322  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2871  013330  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2872  013332  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2873  013334  000000          $CHARCNT: .WORD 0               ;;CHARACTER COUNT STORAGE
2874  013336  000207          $TYPEX: RTS     PC
2875
2876
2877                                  ;ASCII STRING INPUT ROUTINE
2878
2879  013340  017667  000000  000014  .INSTR: MOV     @(SP)+,.MSG     ;PICK UP MESSAGE
2880  013346  062716  000002                  ADD     #2,(SP)         ;JUMP AROUND MESSAGE FOR RTI
2881  013352  105767  166170                  TSTB    $ENV            ;APT CONTROL
2882  013356  001036                          BNE     INSTR2          ;YES NO TYPE
2883  013360  104401          .INST1: TYPE
2884  013362  000000          .MSG:   0
2885  013364  012704  016136                  MOV     #INBUF,R4       ;GET STARTING LOC OF INBUF
2886  013370  012703  000007                  MOV     #7,R3           ;MAX # OF CHARS
2887  013374  105777  166044          1$:     TSTB    @$TKS   ;TTY FLAG
2888  013400  100375                          BPL     1$
2889  013402  117714  166040                  MOVB    @$TKB,(R4)      ;TAKE CHAR
2890  013406  142714  000200                  BICB    #200,(R4)       ;STRIP
2891  013412  121427  000025                  CMPB    (R4),#25        ;IS IT < G>
2892  013416  001760                          BEQ     .INST1
2893  013420  122427  000015                  CMPB    (R4)+,#15       ;CHECK FOR CR
2894  013424  001413                          BEQ     INSTR2
2895  013426  105777  166016          2$:     TSTB    @$TPS   ;TEST FLAG
2896  013432  100375                          BPL     2$
2897  013434  117777  166006  166010          MOVB    @$TKB,@$TPB     ;ECHO CHARACTER
2898  013442  005303                          DEC     R3      ;DID YOU TYPE TOO MANY CHARS ?
2899  013444  001353                          BNE     1$
2900  013446  104401          .INSTE: TYPE
2901  013450  015423                          MQM     ;?
2902  013452  000742                          BR      .INST1  ;RETRY
2903  013454  000002          INSTR2: RTI
2904
2905                                  ;CONVERT ASCII STRING TO OCTAL
2906
2907  013456  011605          .PARAM: MOV     (SP),R5 ;PUT CONTENTS OF SP INTO R5
2908  013460  012567  000162                  MOV     (R5)+,LOLIM     ;PUT LOW LIMIT INTO LOLIM
2909  013464  012567  000160                  MOV     (R5)+,HILIM     ;PUT HIGH LIMIT INTO HILIM
2910  013470  012567  000156                  MOV     (R5)+,DEVADR    ;PUT STORE LOC INTO DEVADR
2911  013474  112567  000154                  MOVB    (R5)+,LOBITS    ;PUT MASK INTO LOBITS
```

```
2912  013500  112567  000151            MOVB    (R5)+,ADRCNT    ;PUT COUNT INTO ADRCNT
2913  013504  010516                    MOV     R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
2914  013506  005005            PARAM1: CLR     R5
2915  013510  012704  016136            MOV     #INBUF,R4
2916  013514  122714  000015            CMPB    #15,(R4)        ;CR ?
2917  013520  001420                    BEQ     PARERR  ;YOU TYPED CR TOO SOON !
2918  013522  121427  000060      1$:   CMPB    (R4),#60        ;LOW LIMIT ASCII 0
2919  013526  002415                    BLT     PARERR
2920  013530  121427  000067            CMPB    (R4),#67        ;HIGH LIMIT ASCII 7
2921  013534  003012                    BGT     PARERR
2922  013536  142714  000060            BICB    #60,(R4)        ;CONVERT TO OCTAL
2923  013542  152405                    BISB    (R4)+,R5        ;STORE AWAY ITS AN OK CHAR
2924  013544  122714  000015            CMPB    #15,(R4)        ;CR ?
2925  013550  001414                    BEQ     LIMITS  ;NOW CHECK FOR HIGH &LOW LIMIT CONDS
2926  013552  006305                    ASL     R5      ;ALLOCATE ROOM FOR NEXT CHAR
2927  013554  006305                    ASL     R5
2928  013556  006305                    ASL     R5
2929  013560  000760                    BR      1$
2930  013562  122714  000015    PARERR: CMPB    #15,(R4)        ;CR?
2931  013566  001003                    BNE     120$
2932  013570  005737  013022            TST     @#RDSW          ;CK SWP USED
2933  013574  001023                    BNE     PARTI
2934  013576  104407            120$:   INSTER  ;RETRY
2935  013600  000742                    BR      PARAM1
2936
2937                                    ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2938
2939  013602  020567  000042    LIMITS: CMP     R5,HILIM
2940  013606  101365                    BHI     PARERR  ;THE # IS TOO HIGH
2941  013610  020567  000032            CMP     R5,LOLIM
2942  013614  103762                    BLO     PARERR  ;THE # IS TOO LOW
2943  013616  136705  000032            BITB    LOBITS,R5       ;TEST BY MASKINGTHE #
2944  013622  001357                    BNE     PARERR
2945
2946                                    ;STORE NUMBER AT SPECIFIED ADDRESS
2947
2948  013624  016704  000022            MOV     DEVADR,R4       ;GET STARTING ADDR OF
2949  013630  010524            1$:     MOV     R5,(R4)+        ;STORE AT THIS ADDR
2950  013632  062705  000002            ADD     #2,R5
2951  013636  105367  000013            DECB    ADRCNT  ;HOW MANY TIMES + 2 ?
2952  013642  001372                    BNE     1$
2953  013644  000002            PARTI:  RTI
2954  013646  000000            LOLIM:  0
2955  013650  000000            HILIM:  0
2956  013652  000000            DEVADR: 0
2957  013654  000000            LOBITS: 0
2958          013655            ADRCNT=LOBITS+1
2959
2960                                    ;SAVE PC OF TEST THAT FAILED AND R0-R5
2961
2962  013656  016667  000004  165242 .SAV05: MOV     4(SP),SAVPC
2963
2964                                    ;SAVE R0-R5
2965
2966  013664  010567  165604    SV05:   MOV     R5,$REG5
2967  013670  010467  165576            MOV     R4,$REG4
```

```
2968  013674  010367  165570                  MOV     R3,$REG3
2969  013700  010267  165562                  MOV     R2,$REG2
2970  013704  010167  165554                  MOV     R1,$REG1
2971  013710  010067  165546                  MOV     R0,$REG0
2972  013714  000002                          RTI
2973
2974                                          ;RESTORE R0-R5
2975
2976  013716  016700  165540          .RES05: MOV     $REG0,R0
2977  013722  016701  165536                  MOV     $REG1,R1
2978  013726  016702  165534                  MOV     $REG2,R2
2979  013732  016703  165532                  MOV     $REG3,R3
2980  013736  016704  165530                  MOV     $REG4,R4
2981  013742  016705  165526                  MOV     $REG5,R5
2982  013746  000002                          RTI
2983
2984                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2985
2986  013750  104401                  .CONVR: TYPE
2987  013752  015427                          MCRLF   ;CR  LF
2988  013754  017601  000000                  MOV     @(SP),R1        ;PICK UP DATA POINTER
2989  013760  062716  000002                  ADD     #2,(SP) ;SET UP SP FOR RTI
2990  013764  012167  000130                  MOV     (R1)+,WRDCNT    ;PICK UP # OF WORDS FROM TABLE
2991  013770  112167  000126          1$:     MOVB    (R1)+,CHRCNT    ;PICK UP # OF CHARS FROM TABLE
2992  013774  112167  000123                  MOVB    (R1)+,SPACNT    ;PICK UP # OF SPACES FROM TABLE
2993  014000  013167  000120                  MOV     @(R1)+,BINWRD   ;PICK UP ADDRESS OF  MSG
2994                                                                  ;FROM TABLE
2995  014004  016704  000114          2$:     MOV     BINWRD,R4       ;SAVE
2996  014010  116705  000106                  MOVB    CHRCNT,R5       ;SAVE
2997  014014  012700  016200                  MOV     #TEMP,R0        ;STARTING ADDRESS OF TEMP BLOCK
2998  014020  010403                  3$:     MOV     R4,R3           ;SAVE
2999  014022  042703  177770                  BIC     #177770,R3      ;CLR OUT UPPER BITS .. SAVE CHAR
3000  014026  062703  000260                  ADD     #260,R3 ;CONVERT TO ASCII
3001  014032  110320                          MOVB    R3,(R0)+        ;STORE AWAY
3002  014034  006204                          ASR     R4      ;SHIFT FOR NEXT #
3003  014036  006204                          ASR     R4      ;DITTO
3004  014040  006204                          ASR     R4      ;DITTO
3005  014042  005305                          DEC     R5      ;DEC CHAR COUNT
3006  014044  001365                          BNE     3$      ;DO IT AGAIN ?
3007  014046  012703  016242                  MOV     #MDATA,R3       ;STARTING ADDRESS OF MDATA BLOCK
3008  014052  114023          4$:     MOVB    -(R0),(R3)+     ;REVERSE THE ORDER OF NUMBERS
3009  014054  105367  000042                  DECB    CHRCNT  ;DEC CHAR COUNT
3010  014060  001374                          BNE     4$      ;DO IT AGAIN ?
3011  014062  105767  000035                  TSTB    SPACNT  ;HOW MANY SPACES ?
3012  014066  001405                          BEQ     6$      ;TYPE # IF BR =0
3013  014070  112723  000240          5$:     MOVB    #240,(R3)+      ;"SPACE" IN ASCII
3014  014074  105367  000023                  DECB    SPACNT  ;DEC # OF SPACE COUNT
3015  014100  001373                          BNE     5$      ;DO IT AGAIN ?
3016  014102  105013          6$:     CLRB    (R3)    ;INSERT "0" FOR TTY OUTPUT ROUTINE
3017  014104  104401                          TYPE
3018  014106  016242                          MDATA   ;THIS MESSAGE
3019  014110  005367  000004                  DEC     WRDCNT  ;HOW MANY #'S ?
3020  014114  001325                          BNE     1$      ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3021  014116  000002                          RTI     ;RETURN TO PROGRAM
3022  014120  000000          WRDCNT: 0
3023  014122  000000          CHRCNT: 0
```

```
3024            014123                  SPACNT=CHRCNT+1
3025  014124    000000                  BINWRD: 0
3026
3027                                            ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3028                                            ;BUFFER TO THE CHARACTERS "N" AND "Y".
3029                                            ;IF THE CHARACTER IS "N" CLEAR THE FLAG
3030                                            ;IF THE CHARACTER IS "Y" SET THE FLAG
3031
3032  014126    017605  000000          .SETFLG: MOV    @(SP),R5
3033  014132    122767  000116  001776          CMPB   #'N,INBUF       ;IS IT "N" ?
3034  014140    001002                        BNE    1$
3035  014142    105015                        CLRB   (R5)    ;000
3036  014144    000406                        BR     2$
3037  014146    122767  000131  001762  1$:   CMPB   #'Y,INBUF       ;IS IT "Y" ?
3038  014154    001005                        BNE    3$
3039  014156    112715  177777                MOVB   #-1,(R5)        ;377
3040  014162    062716  000002          2$:   ADD    #2,(SP)
3041  014166    000002                        RTI
3042  014170    104407                  3$:   INSTER ;RETRY
3043  014172    000755                        BR     .SETFLG
3044                                    .SBTTL  ERROR HANDLER ROUTINE
3045
3046                                    ;;**************************************************************
3047                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3048                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3049                                    ;*AND GO TO SAVIT ON ERROR
3050                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3051                                    ;*SW15=1        HALT ON ERROR
3052                                    ;*SW13=1        INHIBIT ERROR TYPEOUTS
3053                                    ;*SW10=1        BELL ON ERROR
3054                                    ;*SW09=1        LOOP ON ERROR
3055                                    ;*CALL
3056                                    ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3057
3058  014174                            $ERROR:
3059  014174    105267  165203          7$:   INCB   $ERFLG            ;;SET THE ERROR FLAG
3060  014200    001775                        BEQ    7$               ;;DON'T LET THE FLAG GO TO ZERO
3061  014202    016777  165174  165232        MOV    $TSTNM,@DISPLAY  ;;DISPLAY TEST NUMBER AND ERROR FLAG
3062  014210    032777  002000  165222        BIT    #BIT10,@SWR      ;;BELL ON ERROR?
3063  014216    001402                        BEQ    1$               ;;NO - SKIP
3064  014220    104401  001516                TYPE   ,$BELL           ;;RING BELL
3065  014224    005267  165162          1$:   INC    $ERTTL           ;;COUNT THE NUMBER OF ERRORS
3066  014230    011667  165162                MOV    (SP),$ERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
3067  014234    162767  000002  165154        SUB    #2,$ERRPC
3068  014242    117767  165150  165144        MOVB   @$ERRPC,$ITEMB   ;;STRIP AND SAVE THE ERROR ITEM CODE
3069  014250    032777  020000  165162        BIT    #BIT13,@SWR      ;;SKIP TYPEOUT IF SET
3070  014256    001004                        BNE    20$              ;;SKIP TYPEOUTS
3071  014260    004767  000072                JSR    PC,SAVIT         ;;GO TO USER ERROR ROUTINE
3072  014264    104401  001523                TYPE   ,$CRLF
3073  014270                            20$:
3074  014270    122767  000001  165250        CMPB   #APTENV,$ENV     ;;RUNNING IN APT MODE
3075  014276    001007                        BNE    2$               ;;NO,SKIP APT ERROR REPORT
3076  014300    116767  165110  000004        MOVB   $ITEMB,21$       ;;SET ITEM NUMBER AS ERROR NUMBER
3077  014306    004767  000016'               JSR    PC,$ATY4         ;;REPORT FATAL ERROR TO APT
3078  014312       000                  21$:   .BYTE  0
3079  014313       000                         .BYTE  0
```

```
3080  014314  000777                  22$:    BR      22$             ;;APT ERROR LOOP
3081  014316  005777  165116          2$:     TST     @SWR            ;;HALT ON ERROR
3082  014322  100001                          BPL     3$              ;;SKIP IF CONTINUE
3083  014324  000000                          HALT                    ;;HALT ON ERROR!
3084  014326  032777  001000  165104  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
3085  014334  001402                          BEQ     4$              ;;BR IF NO
3086  014336  016716  165046                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
3087  014342  005767  165146          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
3088  014346  001402                          BEQ     5$              ;;BR IF NONE
3089  014350  016716  165140                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
3090  014354                          5$:
3091  014354  000002                          RTI                     ;;RETURN
3092  014356  010067  164546          SAVIT:  MOV     R0,HLD0
3093  014362  010167  164544                  MOV     R1,HLD1
3094  014366  010267  164542                  MOV     R2,HLD2
3095  014372  010367  164540                  MOV     R3,HLD3
3096  014376  010467  164536                  MOV     R4,HLD4
3097  014402  010567  164534                  MOV     R5,HLD5
3098  014406  016767  164770  164530        MOV     $TSTNM,HLD6
3099
3100                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
3101
3102                                  ;;**********************************************************************
3103                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3104                                  ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
3105                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
3106
3107  014414                          $ERRTYP:
3108  014414  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3109  014420  010046                          MOV     R0,-(SP)        ;;SAVE R0
3110  014422  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
3111  014424  153700  001414                  BISB    @#$ITEMB,R0
3112  014430  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
3113                                                                  ;;TYPE THE PC OF THE ERROR
3114  014432  016746  164760                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
3115                                                                  ;;ERROR ADDRESS
3116  014436  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3117  014440  000426                          BR      6$              ;;GET OUT
3118  014442  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
3119  014444  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
3120  014446  006300                  ASL     R0
3121  014450  006300                  ASL     R0
3122  014452  062700  001652                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
3123  014456  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
3124  014462  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
3125  014464  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
3126  014466  000000          2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
3127  014470  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3128  014474  012067  000004          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
3129  014500  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
3130  014502  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
3131  014504  000000          4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
3132  014506  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3133  014512  011000          5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
3134  014514  001004                  BNE     7$              ;;GO TYPE THE DATA
3135  014516  012600          6$:     MOV     (SP)+,R0        ;;RESTORE R0
```

```
3136  014520  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3137  014524  000207                          RTS     PC              ;;RETURN
3138  014526                          7$:
3139  014526  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
3140  014530  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3141  014532  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
3142  014534  001770                          BEQ     6$              ;;BR IF NO
3143  014536  104401  014544                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
3144  014542  000771                          BR      7$              ;;LOOP
3145  014544  020040      000         8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
3146          014550                          .EVEN
3147                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3148
3149                                  ;;**************************************************************
3150                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3151                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3152                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3153                                  ;*CALL:
3154                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3155                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
3156                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3157                                  ;*      .BYTE   M               ;;M=1 OR 0
3158                                  ;*                                      ;;1=TYPE LEADING ZEROS
3159                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
3160                                  ;*
3161                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3162                                  ;*$TYPOS OR $TYPOC
3163                                  ;*CALL:
3164                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3165                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
3166                                  ;*
3167                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3168                                  ;*CALL:
3169                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3170                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
3171
3172  014550  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
3173  014554  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
3174  014562  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
3175  014566  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
3176  014572  000406                          BR      $TYPON
3177  014574  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
3178  014602  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3179  014610  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3180  014616  010346                          MOV     R3,-(SP)        ;;SAVE R3
3181  014620  010446                          MOV     R4,-(SP)        ;;SAVE R4
3182  014622  010546                          MOV     R5,-(SP)        ;;SAVE R5
3183  014624  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
3184  014630  005404                          NEG     R4
3185  014632  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3186  014636  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3187  014642  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3188  014646  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3189  014652  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
3190  014654  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3191  014656  000404                          BR      3$              ;;GO DO MSB
```

```
3192  014660  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
3193  014662  006105                        ROL     R5
3194  014664  006105                        ROL     R5
3195  014666  010503                        MOV     R5,R3
3196  014670  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3197  014672  105367  000076                DECB    $OMODE          ;;TYPE THIS DIGIT?
3198  014676  100016                        BPL     7$              ;;BR IF NO
3199  014700  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
3200  014704  001002                        BNE     4$              ;;TEST FOR 0
3201  014706  005704                        TST     R4              ;;SUPPRESS THIS 0?
3202  014710  001403                        BEQ     5$              ;;BR IF YES
3203  014712  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3204  014714  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3205  014720  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3206  014724  110367  000040                MOVB    R3,8$           ;;SAVE FOR TYPING
3207  014730  104401  014770                TYPE    ,8$             ;;GO TYPE THIS DIGIT
3208  014734  105367  000032        7$:     DECB    $OCNT           ;;COUNT BY 1
3209  014740  003347                        BGT     2$              ;;BR IF MORE TO DO
3210  014742  002402                        BLT     6$              ;;BR IF DONE
3211  014744  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3212  014746  000744                        BR      2$              ;;GO DO THE LAST DIGIT
3213  014750  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
3214  014752  012604                        MOV     (SP)+,R4        ;;RESTORE R4
3215  014754  012603                        MOV     (SP)+,R3        ;;RESTORE R3
3216  014756  016666  000002  000004       MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3217  014764  012616                        MOV     (SP)+,(SP)
3218  014766  000002                        RTI                     ;;RETURN
3219  014770     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3220  014771     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3221  014772     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3222  014773     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3223  014774  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3224                                        ;ENTER HERE ON POWER FAILURE
3225
3226
3227  014776                        $PWRDN:
3228  014776  010046                .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
3229  015000  010146                        MOV     R1,-(SP)
3230  015002  010246                        MOV     R2,-(SP)
3231  015004  010346                        MOV     R3,-(SP)
3232  015006  010446                        MOV     R4,-(SP)
3233  015010  010546                        MOV     R5,-(SP)
3234  015012  016746  163006                MOV     24,-(SP)
3235  015016  010667  164074                MOV     SP,SAVSP                ;SAVE STACK POINTER
3236  015022  012767  015034  162774        MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
3237  015030  000000                        HALT                            ;HALT ON POWER DOWN NORMAL
3238  015032  000777                        BR      .
3239
3240                                        ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3241
3242  015034  016706  164056        RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
3243  015040  012605                        MOV     (SP)+,R5                ;RESTORE R0-R5
3244  015042  012604                        MOV     (SP)+,R4
3245  015044  012603                        MOV     (SP)+,R3
3246  015046  012602                        MOV     (SP)+,R2
3247  015050  012601                        MOV     (SP)+,R1
```

```
3248  015052  012600                          MOV     (SP)+,R0
3249  015054  012767  014776  162742          MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
3250  015062  106427  000340                  MTPS    #340
3251  015066  012706  001100                  MOV     #STACK,SP
3252  015072  005067  001102                  CLR     TEMP
3253  015076  005267  001076                  INC     TEMP
3254  015102  001375                          BNE     .-4
3255  015104  104413                          CONVRT
3256  015106  015130                          PFTAB
3257  015110  104401                          TYPE
3258  015112  015432                          MPFAIL
3259  015114  005067  164263                  CLR     $ERFLG
3260  015120  005067  164272                  CLR     $ERRPC
3261  015124  000177  163754                  JMP     @RETURN
3262  015130  000001                  PFTAB:  1
3263  015132     006     002                  .BYTE   6,2
3264  015134  000207                          RETURN
3265  015136  005015  042012  053125  MTITLE: .ASCIZ  <15><12><12>/DUV11 DZDUS-B TAPE C /<15><12>
3266  015144  030461  042040  042132
3267  015152  051525  041055  052040
3268  015160  050101  020105  020103
3269  015166  005015     000
3270  015171     015  053012  041505  MVECTO: .ASCIZ  <15><12>/VEC ADD-/
3271  015176  040440  042104  000055
3272  015204  005015  051461  020124  MREGAD: .ASCIZ  <15><12>/1ST DEV: REC CSR ADD-/
3273  015212  042504  035126  051040
3274  015220  041505  041440  051123
3275  015226  040440  042104  000055
3276  015234  005015  052515  052114  MMULT:  .ASCIZ  <15><12>/MULT DEV ? (Y OR N)-/
3277  015242  042040  053105  037440
3278  015250  024040  020131  051117
3279  015256  047040  026451     000
3280  015263     015  046012  051501  MLASTD: .ASCIZ  <15><12>/LAST DEV: REC CSR ADDR-/
3281  015270  020124  042504  035126
3282  015276  051040  041505  041440
3283  015304  051123  040440  042104
3284  015312  026522     000
3285  015315     075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
3286  015322  042503  020040     000
3287  015327     015  051412  046105  MCOW:   .ASCIZ  <15><12>/SELECT TO RUN @ACTREG/
3288  015334  041505  020124  047524
3289  015342  051040  047125  040040
3290  015350  041501  051124  043505
3291  015356     000
3292  015357     015  047412  043126  MRANGE: .ASCIZ  <15><12>/OVFLO: RETYPE LAST DEV RXCSR ADDS-/
3293  015364  047514  051072  052105
3294  015372  050131  020105  040514
3295  015400  052123  042040  053105
3296  015406  051040  041530  051123
3297  015414  040440  042104  026523
3298  015422     000
3299  015423     040  037440     000  MQM:    .ASCIZ  /  ?/
3300  015427     015  000012          MCRLF:  .ASCIZ   <15><12>
3301  015432  043120  044501  026114  MPFAIL: .ASCIZ  /PFAIL,  RESTART AT TEST IN PROGRESS/
3302  015440  020040  042522  052123
3303  015446  051101  020124  052101
```

```
3304  015454  052040  051505  020124
3305  015462  047111  050040  047522
3306  015470  051107  051505  000123
3307  015476  005015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE C/
3308  015504  043117  050040  051501
3309  015512  020123  040524  042520
3310  015520  041440     000
3311  015523     015  051012     000  MR:     .ASCIZ  <15><12>/R/
3312  015527     015  052012  051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
3313  015534  020124  041520  000055
3314  015542  005015  047514  045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON  TEST? (Y OR N)-/
3315  015550  047440  020116  052040
3316  015556  051505  037524  024040
3317  015564  020131  051117  047040
3318  015572  026451     000
3319  015575     015  021412  047440  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
3320  015602  020106  054523  041516
3321  015610  041440  040510  051522
3322  015616  051440  046105  041505
3323  015624  042524  020104  020050
3324  015632  020061  051117  031040
3325  015640  026451     000
3326  015643     015  044412  020123  MWIRE6: .ASCIZ  <15><12>/!S SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3327  015650  042523  020103  046530
3328  015656  052111  051440  044527
3329  015664  041524  020110  032505
3330  015672  026465  020062  047111
3331  015700  020077  054450  047440
3332  015706  020122  024516  000055
3333  015714  005015  051511  051440  MWIRE5. .ASCIZ  <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3334  015722  041505  051040  041505
3335  015730  051440  044527  041524
3336  015736  020110  032505  026465
3337  015744  020063  047111  020077
3338  015752  054450  047440  020122
3339  015760  024516  000055
3340  015764  005015  051511  047440  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3341  015772  052120  041440  051114
3342  016000  042440  040516  046102
3343  016006  020105  053523  052111
3344  016014  044103  042440  032465
3345  016022  030455  044440  037516
3346  016030  024040  020131  051117
3347  016036  047040  026451     000
3348  016043     015  005012  031510  MEXTJ:  .ASCIZ  <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3349  016050  032461  041440  047117
3350  016056  042516  052103  051117
3351  016064  047440  020116  024077
3352  016072  020131  051117  047040
3353  016100  026451     000
3354  016103     015  020012  043536  MCNTG:  .ASCIZ  <15><12>/  G  /
3355  016110  020040     000
3356  016113     040  053523  036522  MMSWR:  .ASCIZ  / SWR=   /
3357  016120  020040  000040
3358  016124  020040  047040  053505  MMNEW:  .ASCIZ  /   NEW=  /
3359  016132  020075  000040
```

```
3360                                    .EVEN
3361
3362                                            ;BUFFERS FOR INPUT-OUTPUT
3363
3364  016136  000000                    INBUF:  0
3365          016200                    .=.+40
3366  016200  000000                    TEMP:   0
3367          016242                    .=.+40
3368  016242  000000                    MDATA:  0
3369          016304                    .=.+40
3370                                    .SBTTL  SCOPE HANDLER ROUTINE
3371
3372                                    ;;**********************************************************************
3373                                    ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS.  IT WILL INCREMENT
3374                                    ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG. (DISPLAY<7:0>)
3375                                    ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3376                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3377                                    ;*SW14=1         LOOP ON TEST
3378                                    ;*SW11=1         INHIBIT ITERATIONS
3379                                    ;*SW09=1         LOOP ON ERROR
3380                                    ;*SW08=1         LOOP ON TEST IN SWR<7:0>
3381                                    ;*CALL
3382                                    ;*      SCOPE           ;;SCOPE=IOT
3383
3384  016304                            $SCOPE:
3385
3386                                            ;SCOPE LOOP AND INTERATION HANDLER
3387
3388  016304                            .SCOPE:
3389  016304  004767  174376                    JSR     PC,CKSWR
3390  016310  005067  163102                    CLR     $ERRPC          ;CLEAR LAST ERROR PC
3391  016314  022716  003376                    CMP     #TST1+2,(SP)    ;IS SCOPE AT BEGINING OF TEST 1?
3392  016320  001422                            BEQ     $XTSTR          ;YES NO LOOP.
3393
3394  016322  032777  040000  163110  TTST:     BIT     #BIT14,@SWR     ;THIS CODE IS FOR TESTING FOR BIT 14
3395  016330  001412                            BEQ     1$              ;ON LSI WHICH SYSMAC CANNOT HANDLE
3396  016332  016767  163044  163046            MOV     $TSTNM,$LPADR
3397  016340  000406                            BR      1$
3398  016342  105777  163076                    TSTB    @$TKS           ;KEYBOARD DONE?
3399  016346  100123                            BPL     $OVER           ;BR IF NO
3400  016350  017766  163072  177776            MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
3401  016356  032777  040000  163054  1$:       BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
3402  016364  001114                            BNE     $OVER           ;;YES IF SW14=1
3403                                    ;#####START OF CODE FOR THE XOR TESTER#####
3404  016366  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3405                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3406  016370  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3407  016374  012737  016414  000004            MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
3408  016402  005737  177060                    TST     @#177060        ;;TIME OUT ON XOR?
3409  016406  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3410  016412  000463                            BR      $SVLAD          ;;GO TO THE NEXT TEST
3411  016414  022626                    5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
3412  016416  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3413  016422  000423                            BR      7$              ;;LOOP ON THE PRESENT TEST
3414  016424                            6$: ;#####END OF CODE FOR THE XOR TESTER#####
3415  016424  032777  000400  163006            BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
```

```
3416  016432  001404                        BEQ     2$                      ;;BR IF NO
3417  016434  127767  163000  162740        CMPB    @SWR,$TSTNM             ;;ON THE RIGHT TEST?    SWR<7:0>
3418  016442  001465                        BEQ     $OVER                   ;;BR IF YES
3419  016444  105767  162733          2$:   TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
3420  016450  001421                        BEQ     3$                      ;;BR IF NO
3421  016452  126767  162737  162723        CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
3422  016460  101015                        BHI     3$                      ;;BR IF NO
3423  016462  032777  001000  162750        BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
3424  016470  001404                        BEQ     4$                      ;;BR IF NO
3425  016472  016767  162712  162706  7$:   MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
3426  016500  000446                        BR      $OVER
3427  016502  105067  162675          4$:   CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
3428  016506  005067  163000                CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3429  016512  000415                        BR      1$                      ;;ESCAPE TO THE NEXT TEST
3430  016514  032777  004000  162716  3$:   BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
3431  016522  001011                        BNE     1$                      ;;BR IF YES
3432  016524  005767  163004                TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
3433  016530  001406                        BEQ     1$                      ;;      INHIBIT ITERATIONS
3434  016532  005267  162646                INC     $ICNT                   ;;INCREMENT ITERATION COUNT
3435  016536  026767  162750  162640        CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
3436  016544  002024                        BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
3437  016546  012767  000001  162630  1$:   MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
3438  016554  016767  000056  162730        MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
3439  016562  105267  162614          $SVLAD: INCB  $TSTNM                  ;;COUNT TEST NUMBERS
3440  016566  116767  162610  162736        MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
3441  016574  011667  162606                MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
3442  016600  011667  162604                MOV     (SP),$LPERR             ;;SAVE ERROR LOOP ADDRESS
3443  016604  005067  162704                CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
3444  016610  112767  000001  162577        MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3445  016616  016777  162560  162616  $OVER:  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
3446  016624  016716  162556                MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
3447  016630  000002                  4$:   RTI
3448  016632  001407                  BRW:  1407
3449  016634  000432                  BRX:  432
3450  016636  000005                  $MXCNT: 5                               ;;MAX. NUMBER OF ITERATIONS
3451                                  .SBTTL  TRAP DECODER
3452
3453                                  ;;**************************************************************
3454                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3455                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3456                                  ;*OF THE DESIRED ROUTINE.  THEN USING THE ADDRESS OBTAINED IT WILL
3457                                  ;*GO TO THAT ROUTINE.
3458
3459  016640  010046                  $TRAP:  MOV     R0,-(SP)                ;;SAVE R0
3460  016642  016600  000002                  MOV     2(SP),R0                ;;GET TRAP ADDRESS
3461  016646  005740                          TST     -(R0)                   ;;BACKUP BY 2
3462  016650  111000                          MOVB    (R0),R0                 ;;GET RIGHT BYTE OF TRAP
3463  016652  006300                          ASL     R0                      ;;POSITION FOR INDEXING
3464  016654  016000  016674                  MOV     $TRPAD(R0),R0           ;;INDEX TO TABLE
3465  016660  000200                          RTS     R0                      ;;GO TO ROUTINE
3466
3467
3468                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
3469
3470  016662  011646                  $TRAP2: MOV     (SP),-(SP)              ;;MOVE THE PC DOWN
3471  016664  016666  000004  000002          MOV     4(SP),2(SP)             ;;MOVE THE PSW DOWN
```

```
3472  016672  000002                          RTI                         ;;RESTORE THE PSW
3473
3474                                  .SBTTL  TRAP TABLE
3475
3476                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3477                                  ;*BY THE "TRAP" INSTRUCTION.
3478
3479                                  ;       ROUTINE
3480                                  ;       -------
3481  016674  016662                  $TRPAD: .WORD   $TRAP2
3482  016676  013056                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
3483  016700  014574                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3484  016702  014550                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3485  016704  014610                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
3486
3487
3488  016706  013034                          .SCOP1  ;;CALL=SCOP1    TRAP+5(104405)
3489  016710  013340                          .INSTR  ;;CALL=INSTR    TRAP+6(104406)
3490  016712  013446                          .INSTER ;;CALL=INSTER   TRAP+7(104407)
3491  016714  013456                          .PARAM  ;;CALL=PARAM    TRAP+10(104410)
3492  016716  013656                          .SAV05  ;;CALL=SAV05    TRAP+11(104411)
3493  016720  013716                          .RES05  ;;CALL=RES05    TRAP+12(104412)
3494  016722  013750                          .CONVRT ;;CALL=CONVRT   TRAP+13(104413)
3495  016724  014126                          .SETFLG ;;CALL=SETFLG   TRAP+14(104414)
3496                                  ;*****************************************************************
3497                                          ;UTILITIES
3498                                  ;*****************************************************************
3499
3500                                  ;THIS UTILITY CALCULATES PRIORITY LEVEL
3501  016726  006367  000044          DULEV:  ASL     DUPRT   ;SHIFT LEFT
3502  016732  006367  000040                  ASL     DUPRT   ;
3503  016736  006367  000034                  ASL     DUPRT   ;
3504  016742  006367  000030                  ASL     DUPRT   ;
3505  016746  006367  000024                  ASL     DUPRT   ;
3506  016752  016767  000020  000020         MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
3507  016760  162767  000001  000012         SUB     #1,LESS1        ;CREATE LESS1
3508  016766  042767  000037  000004         BIC     #37,LESS1       ;CLEAR TNZVC
3509  016774  000207                  RTS     PC
3510  016776  000240                  DUPRT:  PR5
3511  017000  000200                  LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
3512
3513                                          ;NEW DU ADDRESSES
3514  017002  016767  000126  162702  DUADDR: MOV     DUBASE,RXCSR    ;XXX0
3515  017010  005267  000120                  INC     DUBASE
3516  017014  016767  000114  162672         MOV     DUBASE,HRXCSR   ;XXX1
3517  017022  005267  000106                  INC     DUBASE
3518  017026  016767  000102  162662         MOV     DUBASE,RXDBUF   ;XXX2
3519  017034  016767  000074  162660         MOV     DUBASE,PARCSR   ;XXX2
3520  017042  005267  000066                  INC     DUBASE
3521  017046  016767  000062  162644         MOV     DUBASE,HRXDBUF  ;XXX3
3522  017054  016767  000054  162642         MOV     DUBASE,HPARCSR  ;XXX3
3523  017062  005267  000046                  INC     DUBASE
3524  017066  016767  000042  162632         MOV     DUBASE,TXCSR    ;XXX4
3525  017074  005267  000034                  INC     DUBASE
3526  017100  016767  000030  162622         MOV     DUBASE,HTXCSR   ;XXX5
3527  017106  005267  000022                  INC     DUBASE
```

```
3528  017112  016767  000016  162612          MOV     DUBASE,TXDBUF   ;XXX6
3529  017120  005267  000010                  INC     DUBASE
3530  017124  016767  000004  162602          MOV     DUBASE,HTXDBUF  ;XXX7
3531  017132  000207                          RTS     PC
3532  017134  000000                  DUBASE: 0
3533
3534                                          ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3535                                          ;INFORMATION CONTAINED IN $TMP1 AND IT IS
3536                                          ;SHIFTED IN BY THE CONTENTS OF SHIFT
3537  017136  042777  040000  162562  RPOKE:  BIC     #MTDATA,@TXCSR
3538  017144  005067  162332                  CLR     $TMP2
3539  017150  006067  162324                  ROR     $TMP1   ;FORCE CARRY
3540  017154  006067  162322                  ROR     $TMP2   ;PICK UP CARRY IN BIT 15
3541  017160  006267  162316                  ASR     $TMP2   ;SHIFT INTO BIT 14
3542  017164  042767  100000  162310          BIC     #BIT15,$TMP2    ;CLR BIT 15
3543  017172  056777  162304  162526          BIS     $TMP2,@TXCSR    ;POKE MAINT DATA
3544  017200  042777  020000  162520          BIC     #CLK,@TXCSR     ;POKE CLK
3545  017206  052777  020000  162512          BIS     #CLK,@TXCSR     ;
3546  017214  005367  161702                  DEC     SHIFT
3547  017220  001346                          BNE     RPOKE
3548  017222  000207                          RTS     PC
3549                                          ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3550  017224  016767  162250  162250  ODD8:   MOV     $TMP1,$TMP2     ;SAVE TEMP1
3551  017232  005067  162246                  CLR     $TMP3
3552  017236  012727  000010                  MOV     #8.,(PC)+
3553  017242  000000                  4$:     0
3554  017244  006067  162232          1$:     ROR     $TMP2
3555  017250  005567  162230                  ADC     $TMP3
3556  017254  005367  177762                  DEC     4$
3557  017260  001371                          BNE     1$
3558  017262  006067  162216                  ROR     $TMP3
3559  017266  103404                          BCS     2$
3560  017270  052767  000400  162202          BIS     #BIT8,$TMP1     ;SET ODD PARITY
3561  017276  000403                          BR      3$
3562  017300  042767  000400  162172  2$:     BIC     #BIT8,$TMP1     ;CLR EVEN PARITY
3563                                          ;$TMP1 NOW HAS ODD PARITY CHARACTER
3564  017306  000207                  3$:     RTS     PC
3565
3566                                  ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3567  017310  016767  162164  162164  EVEN8:  MOV     $TMP1,$TMP2     ;SAVE TEMP1
3568  017316  005067  162162                  CLR     $TMP3
3569  017322  012727  000010                  MOV     #8.,(PC)+
3570  017326  000000                  4$:     0
3571  017330  006067  162146          1$:     ROR     $TMP2
3572  017334  005567  162144                  ADC     $TMP3
3573  017340  005367  177762                  DEC     4$
3574  017344  001371                          BNE     1$
3575  017346  006067  162132                  ROR     $TMP3
3576  017352  103004                          BCC     2$
3577  017354  052767  000400  162116          BIS     #BIT8,$TMP1     ;SET EVEN PARITY
3578  017362  000403                          BR      3$
3579  017364  042767  000400  162106  2$:     BIC     #BIT8,$TMP1     ;CLR ODD PARITY
3580                                          ;$TMP1 NOW HAS EVEN PARITY CHARACTER
3581  017372  000207                  3$:     RTS     PC
3582
3583  017374  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
```

```
 3584                                                ;IN MAIN PART OF THE PROGRAM
 3585  017400  000002                          RTI
 3586          000001                  .END
```

```
AAA     003200            1293#
ABASE = 000000             876     917
ACDW1 = 000000             876     919
ACDW2 = 000000             876     920
ACPUOP= 000000             876     891
ACTREG  001166             735#   1249*   1263*   1264*   1271*   2697    2700    2710
ADDW0 = 000000             876     921
ADDW1 = 000000             876     922
ADDW10= 000000             876     931
ADDW11= 000000             876     932
ADDW12= 000000             876     933
ADDW13= 000000             876     934
ADDW14= 000000             876     935
ADDW15= 000000             876     936
ADDW2 = 000000             876     923
ADDW3 = 000000             876     924
ADDW4 = 000000             876     925
ADDW5 = 000000             876     926
ADDW6 = 000000             876     927
ADDW7 = 000000             876     928
ADDW8 = 000000             876     929
ADDW9 = 000000             876     930
ADEVCT= 000000             876     882
ADEVM = 000000             876     918
ADRCNT= 013655            2912*   2951*   2958#
AENV  = 000000             876     887
AENVM = 000000             876     888
AFATAL= 000000             876     879
AMADR1= 000000             876     904
AMADR2= 000000             876     908
AMADR3= 000000             876     911
AMADR4= 000000             876     914
AMAMS1= 000000             876     898
AMAMS2= 000000             876     906
AMAMS3= 000000             876     909
AMAMS4= 000000             876     912
AMSGAD= 000000             876     884
AMSGLG= 000000             876     885
AMSGTY= 000000             876     878
AMTYP1= 000000             876     899
AMTYP2= 000000             876     907
AMTYP3= 000000             876     910
AMTYP4= 000000             876     913
APASS = 000000             876     881
APRIOR= 000000             876
APTCSU= 000040             536#    2827
APTENV= 000001             536#    2820    3074
APTSIZ= 000200             536#    1168
APTSPO= 000100             536#    2822
ASWREG= 000000             876     889
ATESTN= 000000             876     880
AUNIT = 000000             876     883
AUSWR = 000000             876     890
AVECT1= 000000             876     915
AVECT2= 000000             876     916
BASEAD  001154             730#   1231*   1268*   1269    1275*   1277*   2704*   2716*   2720
```

```
BASEIV  001162            733#   1241*  2705*  2717*  2722   2723*  2724   2725*  2726   2727*  2728   2729*
BBB     003026           1248    1252#
BINWRD  014124           2993*   2995   3025#
BITW  = 002000            802#    995#
BITO  = 000001            665#    778    809    971   1002
BITOO = 000001            655#    665
BITO1 = 000002            654#    664
BITO2 = 000004            653#    663
BITO3 = 000010            652#    662
BITO4 = 000020            651#    661
BITO5 = 000040            650#    660
BITO6 = 000100            649#    659
BITO7 = 000200            648#    658
BITO8 = 000400            647#    657   3415
BITO9 = 001000            646#    656   3084   3423
BIT1  = 000002            664#    777    970
BIT10 = 002000            645#    768    802    961    995   2166   2210   3062
BIT11 = 004000            644#    767    960   3430
BIT12 = 010000            643#    766    783    959    976
BIT13 = 020000            642#    765    782    801    958    975    994   3069
BIT14 = 040000            641#    764    781    800    957    974    993   3394   3401
BIT15 = 100000            640#    763    780    799    956    973    992   3542
BIT2  = 000004            663#    776    969
BIT3  = 000010            662#    775    808    968   1001
BIT4  = 000020            661#    774    807    967   1000
BIT5  = 000040            660#    773    806    966    999
BIT6  = 000100            659#    772    805    965    998
BIT7  = 000200            658#    771    804    964    997
BIT8  = 000400            657#    770    786    803    963    979    996   3560   3562   3577   3579
BIT9  = 001000            656#    769    785    962    978
BPTVEC= 000014            672#
BREAK = 000001            809#   1002#  1359   1411   1463   1515   1567   1619   1671   1710   1749   1788   1826
                         1867    1908   1949   1987   2028   2069   2110   2147   2191   2235   2273   2312   2348
                         2384    2420   2456   2489   2522   2555   2588   2624   2660
BRW     016632           3448#
BRX     016634           3449#
CARDET= 010000            766#    959#
CCC     012612           2696    2719   2736#
CHRCNT  014122           2991*   2996   3009*  3023#  3024
CKSWR   012706           2690    2753   2760#  2792   3389
CLK   = 020000            801#    994#  1359   1365   1366   1411   1417   1418   1463   1469   1470   1515   1521
                         1522    1567   1573   1574   1619   1625   1626   1671   1677   1678   1680   1681   1710
                         1716    1717   1719   1720   1749   1755   1756   1758   1759   1788   1794   1795   1797
                         1798    1826   1832   1833   1835   1836   1867   1873   1874   1876   1877   1908   1914
                         1915    1917   1918   1949   1955   1956   1958   1959   1987   1993   1994   1996   1997
                         2028    2034   2035   2037   2038   2069   2075   2076   2078   2079   2110   2116   2117
                         2119    2120   2147   2153   2154   2156   2157   2191   2197   2198   2200   2201   2235
                         2241    2242   2273   2279   2280   2312   2318   2319   2321   2322   2348   2354   2355
                         2357    2358   2384   2390   2391   2393   2394   2420   2426   2427   2429   2430   2456
                         2462    2463   2489   2495   2496   2522   2528   2529   2555   2561   2562   2588   2594
                         2595    2597   2598   2624   2630   2631   2633   2634   2660   2666   2667   2669   2670
                         3544    3545
CNTLU   012762           1198    2771#
CONVRT= 104413           2693    2773   3255   3494#
COUNT   001124            710#   2324*  2331*  2360*  2367*  2396*  2403*  2432*  2439*  2465*  2472*  2498*  2505*
                         2531*   2538*  2564*  2571*  2600*  2607*  2636*  2643*  2672*  2679*
```

```
CR     = 000015            580#   2866   2876
CRLF   = 000200            581#   2837   2876
CTS    = 020000            765#    958#
DDISP  = 177570            587#    843   1156
DEVADR  013652            2910*   2948   2956#
DEVICE  015315            2694    3285#
DF1     002132            1027    1031   1035   1039   1085#
DH1     002067            1025    1029   1033   1074#
DISPLA  001442             843#   1156*  1164*  1189*  3061*  3445*
DISPRE  000174             684#   1164   1189
DNA    = 100000            799#    992#
DNAINT= 000040             806#    999#
DSC    = 100000            763#    956#
DSINTE= 000040             773#    966#
DSR    = 001000            769#    962#
DSWR   = 177570            586#    842   1155
DTR    = 000002            777#    970#
DT1     002116            1026    1030   1034   1080#
DT4     002126            1038    1083#
DUADDR  017002            1230    2721   3514#
DUBASE  017134            1226    1229   2720*  3514   3515*  3516   3517*  3518   3519   3520*  3521   3522   3523*
                          3524    3525*  3526   3527*  3528   3529*  3530   3532#
DULEV   016726            1288    3501#
DUPRT   016776            1287*   3501*  3502*  3503*  3504*  3505*  3506   3510#
DURIS   001740            1054#   2724*
DURIV   001736            1053#   1237   1240   1241   2722*  2729
DUTIS   001744            1056#   2728*
DUTIV   001742            1055#   2726*
EIGHT  = 006000             794#    987#  1466   1518   1570   1622   1791   1952   2113   2150   2194   2238   2276
                          2423    2558
EMTVEC= 000030             675#   1140*  1141*
EM1     001762            1024    1060#
EM2     002022            1028    1066#
EM3     002043            1032    1069#
EM4     001746            1036    1058#
ERRCNT  001114             703#
ERRVEC= 000004             668#   1153   1154*  1165*  3406   3407*  3409*  3412*
EVEN8   017310            2163    2248   3567#
EVEPAR= 001400             797#    990#  1829   1870   1911   1952   2150   2238
EVPAR  = 000400            786#    979#
FIVE   = 000000            791#    984#  1674   1829   1990   2315   2459   2591
FRMERR= 020000             782#    975#  1687   1726   1765   1804
GNS    = ****** U         3482    3483   3484   3485   3488   3489   3490   3491   3492   3493   3494   3495
HDXEN  = 000010             808#   1001#
HILIM   013650            2909*   2939   2955#
HLD0    001130             712#   1080   3092*
HLD1    001132             713#   1080   3093*
HLD2    001134             714#   3094*
HLD3    001136             715#   3095*
HLD4    001140             716#   3096*
HLD5    001142             717#   3097*
HLD6    001144             718#   3098*
HOLD    001120             708#
HPARCS  001724            1047#   3522*
HRXCSR  001714            1043#   3516*
HRXDBU  001720            1045#   3521*
```

```
HT     = 000011            578#   2835   2876
HTXCSR   001730           1049#  3526*
HTXDBU   001734           1051#  3530*
INBUF    016136           1296   1300   2885   2915   3033   3037   3364#
INIFLG   001172            743#  1199   1202*
INSTER=  104407           1304   2934   3042   3490#
INSTR  = 104406           1221   1232   1242   1253   1278   1294   1307   1311   1315   1319   1330   2775   3489#
INSTR2   013454           2882   2894   2903#
IOTVEC=  000020            673#  1138*  1139*
ISYMOD=  000000            790#   983#  1667   1674   1706   1713   1745   1752   1784   1791   1822   1829   1863
                          1870   1904   1911   1945   1952   1983   1990   2024   2031   2065   2072   2106   2113
                          2143   2150   2187   2194   2308   2315   2344   2351   2380   2387   2416   2423
JMRBY    001153            726#  1207*  1322
KEEPAD   001156            731#  1229*  1231   1275   1277   2716
KEEPIV   001164            734#  1240*  2717
LASTAD   001160            732#  1258   1269   1283
LESS1    017000           3506*  3507*  3508*  3511#
LF     = 000012            579#  2870   2876
LIGHTS   001102            693#  2741*
LIMITS   013602           2925   2939#
LOBITS   013654           2911*  2943   2957#  2958
LOCK     001110            701#  2795
LOKFLG   001174            745#
LOLIM    013646           2908*  2941   2954#
MCNTG    016103           2770   3354#
MCOW     015327           2699   3287#
MCRLF    015427           2987   3300#
MDATA    016242           3007   3018   3368#
MEPASS   015476           2692   3307#
MEXT   = 010000            813#  1006#
MEXTJ    016043           1320   3348#
MINT   = 004000            812#  1005#  1359   1411   1463   1515   1567   1619   1671   1710   1749   1788   1826
                          1867   1908   1949   1987   2028   2069   2110   2147   2191   2235   2273   2312   2348
                          2384   2420   2456   2489   2522   2555   2588   2624   2660
MLASTD   015263           1254   3280#
MLOCK    015542           3314#
MMNEW    016124           2775   3358#
MMSWR    016113           2772   3356#
MMULT    015234           1243   3276#
MPFAIL   015432           3258   3301#
MQM      015423           2901   3299#
MR       015523           1341   3311#
MRANGE   015357           1279   3292#
MREGAD   015204           1222   3272#
MRESET=  000400            803#   996#  1354   1356   1406   1408   1458   1460   1510   1512   1562   1564   1614
                          1616   1666   1668   1705   1707   1744   1746   1783   1785   1821   1823   1862   1864
                          1903   1905   1944   1946   1982   1984   2023   2025   2064   2066   2105   2107   2142
                          2144   2186   2188   2230   2232   2268   2270   2307   2309   2343   2345   2379   2381
                          2415   2417   2451   2453   2484   2486   2517   2519   2550   2552   2583   2585   2619
                          2621   2655   2657
MSYNC    015575           1295   3319#
MTDATA=  040000            800#   993#  1359   1411   1463   1515   1567   1619   1671   1710   1749   1788   1826
                          1867   1908   1949   1987   2028   2069   2110   2147   2191   2235   2273   2312   2348
                          2384   2420   2456   2489   2522   2555   2588   2624   2660   3537
MTITLE   015136           1201   3265#
MTSTPC   015527           1331   3312#
```

```
MULTD   001152            725#  1245  1246  2695
MVECTO  015171           1233  3270#
MWIRE4  015764           1316  3340#
MWIRE5  015714           1312  3333#
MWIRE6  015643           1308  3326#
NEXT    001106            700#
NOPAR = 000000            795#   988#  1362  1414  1466  1518  1570  1622  1674  1713  1752  1791  2315
                         2351  2387  2423  2459  2492  2525  2558  2591  2627  2663
ODDPAR= 001000            796#   989#  1990  2031  2072  2113  2194  2276
ODD8    017224           2207  2286  3550#
ONCE    002610           1200  1203#
OPTCLR  001151            724#  1318
OUT     013014           2761  2763  2765  2769  2781#
OUTCRY  012604           2693  2732#
OUTMUL  003166           1251  1276  1287#
OVRRUN= 040000            781#   974#
PARAM = 104410           1223  1234  1255  1280  1332  2776  3491#
PARAM1  013506           2914#  2935
PARCSR  001722           1046#  1355*  1362*  1407*  1414*  1459*  1466*  1511*  1518*  1563*  1570*  1615*  1622*
                         1667*  1674*  1706*  1713*  1745*  1752*  1784*  1791*  1822*  1829*  1863*  1870*  1904*
                         1911*  1945*  1952*  1983*  1990*  2024*  2031*  2065*  2072*  2106*  2113*  2143*  2150*
                         2187*  2194*  2231*  2238*  2269*  2276*  2308*  2315*  2344*  2351*  2380*  2387*  2416*
                         2423*  2452*  2459*  2485*  2492*  2518*  2525*  2551*  2558*  2584*  2591*  2620*  2627*
                         2656*  2663*  3519*
PAREN = 001000            785#   978#
PARER = 010000            783#   976#  1838  1879  1920  1961  1999  2040  2081  2122
PARERR  013562           2917  2919  2921  2930#  2940  2942  2944
PARTI   013644           2933  2953#
PASCNT  001112            702#  1176*  2740*  2741  2742
PFTAB   015130           3256  3262#
PIRQ  = 177772            585#
PIRQVE= 000240            679#
POPRO = 012600            758#   951#
POP1SP= 005726            756#   949#
POP2SP= 022626            760#   953#
PR0   = 000000            602#
PR1   = 000040            603#
PR2   = 000100            604#
PR3   = 000140            605#
PR4   = 000200            606#  3511
PR5   = 000240            607#  3510
PR6   = 000300            608#
PR7   = 000340            609#
PS    = 177776            582#   583
PSW   = 177776            583#
PUSHR0= 010046            757#   950#
PUSH1S= 005746            755#   948#
PUSH2S= 024646            759#   952#
PWRVEC= 000024            674#  1144*  1145*
RDSW    013022           2771*  2781*  2783#  2932
RECACT= 004000            767#   960#
REPLAY  012510           2712  2718  2720#
RESTAR  015034           3236  3242#
RESTRT  012670           2713  2744  2751#
RESVEC= 000010            669#
RESO5 = 104412           3493#
```

```
RETURN  001104            699# 1181* 1338* 1340* 1342  3261
RING  = 040000            764# 957#
RINTEN= 000100            772# 965#
ROTADD  001170            738# 1250* 1262* 1264  1266* 1271  1274* 2707* 2710  2714*
RPOKE   017136           1371  1383  1387  1423  1435  1439  1475  1487  1491  1527  1539  1543  1579
                         1591  1595  1631  1643  1647  1685  1724  1763  1802  1841  1882  1923  1964
                         2002  2043  2084  2125  2168  2212  2250  2288  2327  2363  2399  2435  2468
                         2501  2534  2567  2603  2639  2675  3537# 3547
RTS   = 000004            776# 969#
RUNA  = ****** U            1   51   140   513
RUNB  = ****** U            1   51   140   513
RUNC  = 000000              1   51   140   513
RUND  = ****** U           21   56   143   518
RUNE  = ****** U           21   56   143   518
RUNF  = ****** U           21   56   143   518
RUNIT   012414           2698  2704# 2711
RXCSR   001712           1042# 1363* 1372  1415* 1424  1467* 1476  1519* 1528  1571* 1580  1623* 1632
                         1675* 1714* 1753* 1792* 1830* 1842  1871* 1883  1912* 1924  1953* 1965  1991*
                         2003  2032* 2044  2073* 2085  2114* 2126  2151* 2195* 2239* 2277* 2316* 2323*
                         2328  2352* 2359* 2364  2388* 2395* 2400  2424* 2431* 2436  2460* 2464* 2469
                         2493* 2497* 2502  2526* 2530* 2535  2559* 2563* 2568  2592* 2599* 2604  2628*
                         2635* 2640  2664* 2671* 2676  2734  3514*
RXDBUF  001716           1044# 1367  1375  1390  1419  1427  1442  1471  1479  1494  1523  1531  1546
                         1575  1583  1598  1627  1635  1650  1686  1688  1725  1727  1764  1766  1803
                         1805  1837  1845  1878  1886  1919  1927  1960  1968  1998  2006  2039  2047
                         2080  2088  2121  2129  2158  2169  2202  2213  2243  2251  2281  2289  3518*
RXDONE= 000200            771# 964#
RXERR = 100000            780# 973#  1687  1726  1765  1804  1838  1879  1920  1961  1999  2040  2081
                         2122
SAVIT   014356           3071  3092#
SAVPC   001126            711# 2962*
SAVSP   001116            704# 3235*  3242
SAVO5 = 104411           3492#
SCOP1 = 104405           3488#
SEND  = 000020            807# 1000#
SEREC   001150            723# 1314
SETFLG= 104414           1244  1309  1313  1317  1321  3495#
SEVEN = 004000            793#  986#  1362  1414  1752  1911  2072  2387  2525  2663
SEXMIT  001147            722# 1310
SHIFT   001122            709# 1369* 1381* 1385* 1421* 1433* 1437* 1473* 1485* 1489* 1525* 1537* 1541*
                         1577* 1589* 1593* 1629* 1641* 1645* 1682* 1721* 1760* 1799* 1839* 1880* 1921*
                         1962* 2000* 2041* 2082* 2123* 2161* 2205* 2246* 2284* 2326* 2362* 2398* 2434*
                         2467* 2500* 2533* 2566* 2602* 2638* 2674* 3546*
SIX   = 002000            792#  985#  1713  1870  2031  2351  2492  2627
SPACNT= 014123           2992* 3011  3014* 3024#
SRD   = 002000            768#  961#
STACK = 001100            573# 1136  1172  1326  3251
STD   = 000010            775#  968#
STFLG   001173            744# 1175*
STKLMT= 177774            584#
STPSYN= 000400            770#  963#  2323  2359  2395  2431  2464  2497  2530  2563  2599  2635  2671
SVO5    013664           2966#
SWR     001440            842# 1134  1155* 1157  1163* 1170* 1186  1188* 1193  1209  1328  2762  2793
                         3062  3069  3081  3084  3394  3401  3415  3417  3423  3430
SWREG   000176            685# 1163  1188  1193  2762  2779  2786
SWREGL  013024           2774  2784#
```

```
SW0    = 000001            637#
SW00   = 000001            627#    637     1209
SW01   = 000002            626#    636     1328
SW02   = 000004            625#    635
SW03   = 000010            624#    634
SW04   = 000020            623#    633
SW05   = 000040            622#    632
SW06   = 000100            621#    631
SW07   = 000200            620#    630
SW08   = 000400            619#    629
SW09   = 001000            618#    628     2793
SW1    = 000002            636#
SW10   = 002000            617#
SW11   = 004000            616#
SW12   = 010000            615#
SW13   = 020000            614#
SW14   = 040000            613#
SW15   = 100000            612#
SW2    = 000004            635#
SW3    = 000010            634#
SW4    = 000020            633#
SW5    = 000040            632#
SW6    = 000100            631#
SW7    = 000200            630#
SW8    = 000400            629#
SW9    = 001000            628#
SYNCNO  001146             721#    1298*   1302*
SYNEXT= 020000             789#     982#   1355    1362    1407    1414    1459    1466    1511    1518    1563    1570    1615
                          1622    2231    2238    2269    2276    2452    2459    2485    2492    2518    2525    2551    2558
SYNINT= 030000             788#     981#   2584    2591    2620    2627    2656    2663
SYNSCH= 000020             774#     967#   1363    1415    1467    1519    1571    1623    1675    1714    1753    1792    1830
                          1871    1912    1953    1991    2032    2073    2114    2151    2195    2239    2277    2316    2352
                          2388    2424    2460    2493    2526    2559    2592    2628    2664
SYSTST= 014000             814#    1007#
TBITVE= 000014             670#
TEMP    016200            2997    3252*   3253*   3366#
TKVEC  = 000060            677#
TPVEC  = 000064            678#
TRAPVE= 000034             676#    1142*   1143*
TRPREG  017374            3583#
TRTVEC= 000014             671#
TST1    003374            1333    1340    1353#   2752    3391
TST10   005230            1704#
TST11   005370            1743#
TST12   005530            1782#
TST13   005670            1820#
TST14   006040            1861#
TST15   006210            1902#
TST16   006360            1943#
TST17   006530            1981#
TST2    003606            1405#
TST20   006700            2022#
TST21   007050            2063#
TST22   007220            2104#
TST23   007370            2141#
TST24   007554            2185#
```

```
TST25   007740          2229#
TST26   010074          2267#
TST27   010230          2306#
TST3    004020          1457#
TST30   010400          2342#
TST31   010550          2378#
TST32   010720          2414#
TST33   011070          2450#
TST34   011224          2483#
TST35   011360          2516#
TST36   011514          2549#
TST37   011650          2582#
TST4    004232          1509#
TST40   012020          2618#
TST41   012170          2654#
TST5    004444          1561#
TST6    004656          1613#
TST7    005070          1665#
TTST    016322          3394#
TXCSR   001726          1048#  1354*  1356*  1359*  1365*  1366*  1406*  1408*  1411*  1417*  1418*  1458*  1460*
                        1463*  1469*  1470*  1510*  1512*  1515*  1521*  1522*  1562*  1564*  1567*  1573*  1574*
                        1614*  1616*  1619*  1625*  1626*  1666*  1668*  1671*  1677*  1678*  1680*  1681*  1705*
                        1707*  1710*  1716*  1717*  1719*  1720*  1744*  1746*  1749*  1755*  1756*  1758*  1759*
                        1783*  1785*  1788*  1794*  1795*  1797*  1798*  1821*  1823*  1826*  1832*  1833*  1835*
                        1836*  1862*  1864*  1867*  1873*  1874*  1876*  1877*  1903*  1905*  1908*  1914*  1915*
                        1917*  1918*  1944*  1946*  1949*  1955*  1956*  1958*  1959*  1982*  1984*  1987*  1993*
                        1994*  1996*  1997*  2023*  2025*  2028*  2034*  2035*  2037*  2038*  2064*  2066*  2069*
                        2075*  2076*  2078*  2079*  2105*  2107*  2110*  2116*  2117*  2119*  2120*  2142*  2144*
                        2147*  2153*  2154*  2156*  2157*  2186*  2188*  2191*  2197*  2198*  2200*  2201*  2230*
                        2232*  2235*  2241*  2242*  2268*  2270*  2273*  2279*  2280*  2307*  2309*  2312*  2318*
                        2319*  2321*  2322*  2343*  2345*  2348*  2354*  2355*  2357*  2358*  2379*  2381*  2384*
                        2390*  2391*  2393*  2394*  2415*  2417*  2420*  2426*  2427*  2429*  2430*  2451*  2453*
                        2456*  2462*  2463*  2484*  2486*  2489*  2495*  2496*  2517*  2519*  2522*  2528*  2529*
                        2550*  2552*  2555*  2561*  2562*  2583*  2585*  2588*  2594*  2595*  2597*  2598*  2619*
                        2621*  2624*  2630*  2631*  2633*  2634*  2655*  2657*  2660*  2666*  2667*  2669*  2670*
                        3524*  3537*  3543*  3544*  3545*
TXDBUF  001732          1050#  3528*
TXDONE= 000200           804#   997#
TXINTE= 000100           805#   998#
TYPE   = 104401         1201   1341   2691   2694   2699   2770   2772   2840   2883   2900   2986   3017   3064
                        3072   3108   3125   3127   3130   3132   3136   3143   3207   3257   3482#
TYPOC  = 104402         3116   3140   3483#
TYPON  = 104404         3485#
TYPOS  = 104403         3484#
USER   = 000000          811#   1004#
VOID   = 000001          778#   971#
WRDCNT  014120          2990*  3019*  3022#
ZERO    012422          2705#
$APTHD  002136          1107   1113#
$ASTAT= ****** U         536
$ATYC   000024R          536#
$ATY1   000000R          536#
$ATY3   000006R          536#   2825
$ATY4   000016R          536#   3077
$AUTOB  001434           839#
$BASE   001602           917#
```

```
$BDADR  001422            834#
$BDDAT  001426            836#
$BELL   001516            868#   3064    3092
$CDW1   001606            919#
$CDW2   001610            920#
$CHARC  013334           2842*   2852*   2859    2868*   2873#
$CKSWR= ****** U          3488
$CMTAG  001400            822#   1131    1132    1140    1146    1147
$CM1  = 000006            854#    855#    856#    857#    858#    859#    860#
$CM2  = 000014            854#    855#    856#    857#    858#    859#    860#
$CM3  = 000006            852#    854
$CM4  = 000006            860#    861#    862#    863#    864#    865#    866#
$CPUOP  001554            891#
$CRLF   001523            870#   2841    2876    3072    3092    3108    3127    3132    3136
$DDW0   001612            921#
$DDW1   001614            922#
$DDW10  001636            931#
$DDW11  001640            932#
$DDW12  001642            933#
$DDW13  001644            934#
$DDW14  001646            935#
$DDW15  001650            936#
$DDW2   001616            923#
$DDW3   001620            924#
$DDW4   001622            925#
$DDW5   001624            926#
$DDW6   001626            927#
$DDW7   001630            928#
$DDW8   001632            929#
$DDW9   001634            930#
$DEVCT  001536            882#
$DEVM   001604            918#
$E    = 000002            534#
$ENDAD  012660           1094    2747#
$ENV    001546            536     887#   1203    2820    2881    3074
$ENVM   001547            536     888#   1168    2822    2827
$ERFLG  001403            825#   1177*   2739*   3059*   3092    3259*   3375    3419    3421    3427*   3451
$ERMAX  001415            831#   1148*   3421    3444*   3451
$ERROR  014174           1140    3058#
$ERRPC  001416            832#   1080    1083    1179*   2738*   3066*   3067*   3068    3092    3114    3260*   3390*
$ERRTB  001652           1022#   3122
$ERRTY  014414           3107#
$ERTTL  001412            829#   1178*   3065*   3092
$ESCAP  001514            867#   1147*   3087    3089    3092    3443*
$ETABL  001546            886#
$ETEND  001652            939#   1119
$FATAL  001530            536*    879#
$FFLG   000244R           536#*
$FILLC  001456            850#   2845    2876
$FILLS  001455            849#   2876
$GDADR  001420            833#
$GDDAT  001424            835#
$GTSWR= ****** U          3487
$HIBTS  002136           1114#
$ICNT   001404            826#   3434*   3435    3437*   3450
$INTAG  001435            840#
```

```
$ITEMB  001414            830#   3068*  3076   3092   3111
$LF     001524            871#   2876   3092
$LFLG   000243R           536#*
$LPADR  001406            827#   1149*  2752*  3396*  3425*  3441*  3446   3450
$LPERR  001410            828#   1150*  3086   3425   3442*  3450
$MADR1  001560            904#
$MADR2  001564            908#
$MADR3  001570            911#
$MADR4  001574            914#
$MAIL   001526            877#   1115   1119   1167   2820   3074   3440
$MAMS1  001556            898#
$MAMS2  001562            906#
$MAMS3  001566            909#
$MAMS4  001572            912#
$MBADR  002140           1115#
$MFLG   000242R           536#*
$MSGAD  001542            536*    884#
$MSGLG  001544            536*    885#
$MSGTY  001526            536*    878#
$MTYP1  001557            899#
$MTYP2  001563            907#
$MTYP3  001567            910#
$MTYP4  001573            913#
$MXCNT  016636           3438    3450#
$N    = 000000            534#   2683#
$NULL   001454            848#   2847   2876
$NWTST= 000000           1352#   1404#  1456#  1508#  1560#  1612#  1664#  1703#  1742#  1781#  1819#  1860#  1901#
                         1942#   1980#  2021#  2062#  2103#  2140#  2184#  2228#  2266#  2305#  2341#  2377#  2413#
                         2449#   2482#  2515#  2548#  2581#  2617#  2653#
$OCNT   014772           3179*   3208*  3221#
$OMODE  014774           3174*   3178*  3183   3186*  3197*  3223#
$OVER   016616           3399    3402   3418   3426   3436   3445#
$PASS   001534            881#   1167*  2742*  3432   3451
$PASTM  002144           1117#
$PWRDN  014776           1144    3227#
$QUES   001522            869#   2876   3092
$RDCHR= ****** U         3488
$RDDEC= ****** U         3488
$RDLIN= ****** U         3488
$RDOCT= ****** U         3488
$REGAD  001460            852#
$REG0   001462            854#   2971*  2976
$REG1   001464            855#   2970*  2977
$REG2   001466            856#   2969*  2978
$REG3   001470            857#   2968*  2979
$REG4   001472            858#   2967*  2980
$REG5   001474            859#   2966*  2981
$R2A  = ****** U         3488
$SAVRE= ****** U         3488
$SCOPE  016304           1138    3384#
$SETUP= 000017           1120#   1137   1138   1140   1142   1144   1146   1147   1149   3059   3084   3091   3385
$STUP = 177777           1120#
$SVLAD  016562           3410    3439#
$SVPC = 002136           1092#   1097
$SWR  = 177400            525#    866    867    868   1146   1147   1149   1150   1354   1406   1458   1510   1562
                         1614    1666   1705   1744   1783   1821   1862   1903   1944   1982   2023   2064   2105
```

```
                        2142    2186    2230    2268    2307    2343    2379    2415    2451    2484    2517    2550    2583
                        2619    2655    3050    3051    3052    3053    3054    3062    3069    3081    3084    3092    3376
                        3377    3378    3379    3380    3401    3413    3415    3416    3419    3420    3421    3428    3429
                        3430    3442    3445    3450
$SWREG  001550           889#   1170
$SWRMK= 000000          3380    3381    3417
$TESTN  001532           880#   3440*
$TIMES  001512           866#   1146*   3428*   3435    3438*   3450
$TKB    001446           845#   2766    2889    2897    3400
$TKS    001444           844#   2764    2887    3398
$TMP0   001476           860#
$TMP1   001500           861#   1370*   1382*   1386*   1422*   1434*   1438*   1474*   1486*   1490*   1526*   1538*   1542*
                        1578*   1590*   1594*   1630*   1642*   1646*   1683*   1722*   1761*   1800*   1840*   1881*   1922*
                        1963*   2001*   2042*   2083*   2124*   2162*   2165*   2166*   2206*   2209*   2210*   2247*   2285*
                        2325*   2361*   2397*   2433*   2466*   2499*   2532*   2565*   2601*   2637*   2673*   3539*   3550
                        3560*   3562*   3567    3577*   3579*
$TMP2   001502           862#   3538*   3540*   3541*   3542*   3543    3550*   3554*   3567*   3571*
$TMP3   001504           863#   3551*   3555*   3558*   3568*   3572*   3575*
$TMP4   001506           864#
$TMP5   001510           865#
$TN   = 000042           536#   1352    1354#   1404    1406#   1456    1458#   1508    1510#   1560    1562#   1612    1614#
                        1664    1666#   1703    1705#   1742    1744#   1781    1783#   1819    1821#   1860    1862#   1901
                        1903#   1942    1944#   1980    1982#   2021    2023#   2062    2064#   2103    2105#   2140    2142#
                        2184    2186#   2228    2230#   2266    2268#   2305    2307#   2341    2343#   2377    2379#   2413
                        2415#   2449    2451#   2482    2484#   2515    2517#   2548    2550#   2581    2583#   2617    2619#
                        2653    2655#
$TPB    001452           847#   2865*   2876    2897*
$TPFLG  001457           851#   2814    2876
$TPS    001450           846#   2863    2876    2895
$TRAP   016640          1142    3459#
$TRAP2  016662          3470#   3481
$TRP  = 000015          3474#   3483#   3484#   3485#   3486#   3488    3489#   3490#   3491#   3492#   3493#   3494#   3495#
                        3496#
$TRPAD  016674          3464    3481#
$TSTM   002142          1116#
$TSTNM  001402           824#   1180*   1335    1338    2737*   3061    3092    3098    3375    3396    3417    3439*   3440
                        3445    3451
$TYPBN= ****** U        3486
$TYPDS= ****** U        3486
$TYPE   013056           536    2814#   3474    3482
$TYPEC  013270          2844    2851    2858    2863#   2864
$TYPEX  013336          2869    2871    2874#
$TYPOC  014574          3177#   3483
$TYPON  014610          3176    3179#   3485
$TYPOS  014550          3172#   3484
$UNIT   001540           883#
$UNITM  002146          1118#
$USWR   001552           890#   1205
$VECT1  001576           915#
$VECT2  001600           916#
$XTSTR  016366          3392    3404#
$OFILL  014773          3173*   3177*   3187    3222#
$40CAT= ****** U        3071    3401
.     = 017402           536#    570#    683#    686#    691#    746#    747#    821#    872    1078#   1092    1093#   1095#
                        1097#   1103    1104#   1106#   1108#   1135    1149    1150    1373    1377    1392    1425    1429
                        1444    1477    1481    1496    1529    1533    1548    1581    1585    1600    1633    1637    1652
```

```
                        1690    1729    1768    1807    1843    1847    1884    1888    1925    1929    1966    1970    2004
                        2008    2045    2049    2086    2090    2127    2131    2171    2215    2253    2291    2329    2365
                        2401    2437    2470    2503    2536    2569    2605    2641    2677    2876    3092    3146#   3238
                        3254    3365#   3367#   3369#   3450    3451
.BEGIN  003310          1197    1208    1211    1326#   2754
.CONVR  013750          2986#   3494
.EOP    012340          2689#
.INSTE  013446          2900#   3490
.INSTR  013340          2879#   3489
.INST1  013360          2883#   2892    2902
.MSG    013362          2766*   2767*   2768    2879*   2884#
.PARAM  013456          2907#   3491
.PFAIL  014776          1174    3228#   3249
.RESO5  013716          2976#   3493
.SAVO5  013656          2962#   3492
.SCOPE  016304          3388#
.SCOP1  013034          2792#   3488
.SETFL  014126          3032#   3043    3495
.START  002152           687    1129#   1181    2703
.$ASTA= ****** U         536
.$X   = 002136          1103#   1108
```

```
COMMEN    680#
ENDCOM    680#
ERROR     574#  1374  1378  1393  1426  1430  1445  1478  1482  1497  1530  1534  1549  1582  1586
         1601  1634  1638  1653  1691  1730  1769  1808  1844  1848  1885  1889  1926  1930  1967
         1971  2005  2009  2046  2050  2087  2091  2128  2132  2172  2216  2254  2292  2330  2366
         2402  2438  2471  2504  2537  2570  2606  2642  2678
ESCAPE    680#
GETPRI    680#
GETSWR    680#
MULT      680#
NEWTST    680#  1352  1404  1456  1508  1560  1612  1664  1703  1742  1781  1819  1860  1901  1942
         1980  2021  2062  2103  2140  2184  2228  2266  2305  2341  2377  2413  2449  2482  2515
         2548  2581  2617  2653
POP       536   680#
PRGEND    558#  2682
PRGFRT    537#   559
PUSH      536   680#
PUSSYF    555#
REPORT    525#   680#
RSETUP    542#  1354  1406  1458  1510  1562  1614  1666  1705  1744  1783  1821  1862  1903  1944
         1982  2023  2064  2105  2142  2186  2230  2268  2307  2343  2379  2415  2451  2484  2517
         2550  2583  2619  2655
SC        534#  3385
SCOPE     575#  1353  1405  1457  1509  1561  1613  1665  1704  1743  1782  1820  1861  1902  1943
         1981  2022  2063  2104  2141  2185  2229  2267  2306  2342  2378  2414  2450  2483  2516
         2549  2582  2618  2654  2689
SC1       534#  3447
SETPRI    680#
SETTRA    525#  3474#  3483  3484  3485  3488  3489  3490  3491  3492  3493  3494  3495
SETUP     680#  1129
SKIP      680#
SLASH     680#
SPACE     680#
STARS     536   680#   817   872   875  1090  1100  1102  1109  1352  1404  1456  1508  1560  1612
         1664  1703  1742  1781  1819  1860  1901  1942  1980  2021  2062  2103  2140  2184  2228
         2266  2305  2341  2377  2413  2449  2482  2515  2548  2581  2617  2653  2799  3046  3102
         3149  3372  3453
SWRSU     680#  1151#
TRMTRP   3474#
TSETUP    543#
TYPBIN    680#
TYPDEC    680#
TYPNAM    680#
TYPNUM    680#
TYPOCS    680#
TYPOCT    680#  3114  3138
TYPTXT    680#
$BEGIN    534#  1323
$BINAR    554#
$BUFFE    536#  3361
$CABLE    556#
$CATCH    528#   570
$CK       535#  2755
$CLRVE    534#  1203
$CONVR    536#  2983
$DNA      552#
```

```
$EOP      535#  2682
$GETFL    534#  1242  1307  1311  1315  1319
$GETPA    534#  1221  1232  1252  1278  1330
$GETSY    534#  1289
$HEADE    525#   559
$INSTR    536#  2876
$ISOB     550#  2134  2178  2222  2260
$MATCH    547#
$MRR      541#
$MRRW     539#
$MRW      540#
$MSG      536#  3265
$NEED     557#  3092
$PARAM    536#  2904
$PFAIL    536#  3224
$POKE     545#
$POKER    546#  1364  1416  1468  1520  1572  1624  1679  1718  1757  1796  1834  1875  1916  1957
          1995  2036  2077  2118  2155  2199  2240  2278  2320  2356  2392  2428  2461  2494  2527
          2560  2596  2632  2668
$RCNET    557#
$RECAC    545#
$REG      536#  2959
$RESET    538#  1354  1356  1406  1408  1458  1460  1510  1512  1562  1564  1614  1616  1666  1668
          1705  1707  1744  1746  1783  1785  1821  1823  1862  1864  1903  1905  1944  1946  1982
          1984  2023  2025  2064  2066  2105  2107  2142  2144  2186  2188  2230  2232  2268  2270
          2307  2309  2343  2345  2379  2381  2415  2417  2451  2453  2484  2486  2517  2519  2550
          2552  2583  2585  2619  2621  2655  2657
$RXACT    544#
$SCOPE    534#  3370
$SCOP1    535#  2789
$SETFL    534#  3026
$SETVE    529#   680
$START    534#  1121
$STRIP    551#  2298  2334  2370  2406  2442  2475  2508  2541  2574  2610  2646
$SYMBO    525#   748   941
$SYNCR    547#  1675  1714  1753  1792  1830  1871  1912  1953  1991  2032  2073  2114  2151  2195
          2316  2352  2388  2424  2592  2628  2664
$TRPAR    553#
$TSTNO    534#  1352  1404  1456  1508  1560  1612  1664  1703  1742  1781  1819  1860  1901  1942
          1980  2021  2062  2103  2140  2184  2228  2266  2305  2341  2377  2413  2449  2482  2515
          2548  2581  2617  2653
$UNIBU    537#
$VARIA    532#   690
$WORDF    548#  1656  1695  1734  1773
$WORDO    547#  1344  1396  1448  1500  1552  1604
$WORDP    549#  1812  1853  1894  1935  1973  2014  2055  2096
$$CMRE    815#   854   855   856   857   858   859
$$CMTM    815#   860   861   862   863   864   865
$$ESCA    680#
$$NEWT    680#  1352  1404  1456  1508  1560  1612  1664  1703  1742  1781  1819  1860  1901  1942
          1980  2021  2062  2103  2140  2184  2228  2266  2305  2341  2377  2413  2449  2482  2515
          2548  2581  2617  2653
$$SET     3474# 3483  3484  3485  3488  3489  3490  3491  3492  3493  3494  3495
$$SETM    1167#
$$SKIP    680#
.EQUAT    525#   570
```

```
.HEADE    525#
.SETUP    525#   1120
.$ACT1    525#   1088
.$APTB    525#    873#
.$APTH    525#   1098
.$APTY    525#    536
.$CATC    525#
.$CMTA    525#    815
.$EOP     525#
.$ERRO    525#   3044
.$ERRT    525#   3100
.$POWE    525#
.$SCOP    525#   3370
.$TRAP    525#   3451
.$TYPE    525#   2797
.$TYPO    525#   3147


. ABS.   017402     000


 ERRORS DETECTED:  0

 DZDUSB,DZDUSB/SOL/CRF=DZDUS1/EQ:RUNC,DZDUS2,DZDUSB
 RUN-TIME:  22 12 1 SECONDS
 RUN-TIME RATIO:  319/36=8.€
 CORE USED:  30K  (59 PAGES)
```