RH70/RS03

MAINT MODE DIAGNOSTIC
MD-11-DERSC-B

EP-DERSC-B-DL-A

FICHE 1 OF 1

NOV 1976
digital
MADE IN US

```
                        .REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DERSC-B-D
PRODUCT NAME:	RH11-RS03 MAINTENANCE MODE DIAGNOSTIC
DATE CREATED:	AUGUST-1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	STANLEY HARACKIEWICZ

CONTENTS

DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WILL LET THE OPERATOR SELECT ONE OF TWO MODES OF OPERATION. THE OPERATOR MAY SELECT WHICH DRIVE HE WANTS TESTED OR HE CAN LET THE PROGRAM SEQUENCE THROUGH ALL THE DRIVES ON THE SYSTEM.

THE FIRST PART OF THIS DIAGNOSTIC WILL TEST THE DRIVE REGISTERS ASSOCIATED WITH THE DRIVE UNDER TEST. THE PROGRAM WILL ALSO TEST THE RH CONTROLLER REGISTERS TO CONFIRM THAT, FOR THE MOST PART, THE CONTROLLER IS WORKING CORRECTLY.

THE SECOND PART OF THIS DIAGNOSTIC WILL TEST THE DRIVE IN "MAINTENANCE MODE".

THE RS03 HAS BEEN DESIGNED WITH BUILT-IN TEST CAPABILITIES. THIS "MAINTENANCE MODE" TEST CAPABILITY ISOLATES THE DIGITAL ELECTRONICS FROM THE ANALOG AND ALLOWS INDEPENDENT TESTING OF THE DIGITAL LOGIC. THEREFORE, FAILURES LOCATED ENTIRELY IN THE LOGIC CAN BE SEPARATED FROM FAILURES OCCURRING IN THE ANALOG ELECTRONICS OR THE HEAD/DISK SUBASSEMBLY.

1.1 DESIGN PHILOSOPHY

BY SETTING BIT 00 IN THE MAINTENANCE REGISTER, THE MAINTENANCE MODE LOGIC IS ENABLED, AND THE REMAINING READ/WRITE BITS IN THE MAINTENANCE REGISTER ARE SUBSTITUTED FOR THE CORRESPONDING SIGNALS NORMALLY ORIGINATING FROM THE HEAD/DISK SUBASSEMBLY. THE READ-ONLY BITS IN THE MAINTENANCE REGISTER REFLECT THE STATES OF MAJOR SIGNALS DURING DRIVE OPERATION. BY SETTING AND CLEARING THE READ/WRITE BITS IN PREDETERMINED SEQUENCES AND SIMULTANEOUSLY MONITORING THE READ-ONLY BITS, IT IS POSSIBLE TO VERIFY THE OPERATION OF ALL OF THE DRIVE'S LOGIC. THIS INCLUDES ALL DRIVE TIMING AS WELL AS THE LOGIC ASSOCIATED WITH READING AND WRITING DATA.

--CAUTION--

A THOROUGH UNDERSTANDING OF THE RS03 LOGIC IS REQUIRED TO UTILIZE THIS DIAGNOSTIC EFFECTIVELY. REFER TO SECTIONS 2 AND 3 OF THE "RS03 DECDISK SERVICE MANUAL" (DEC-00-HRS3A-A-D) FOR DESCRIPTIONS OF THE DRIVE LOGIC.

2. REQUIREMENTS

2.1 EQUIPMENT

```
156
157
```

PDP-11 WITH A MINIMUM OF 8K OF MEMORY AND AN RH11 CONTROLLER WITH A
RS03 DISK.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTR.. SWITCH SETTINGS

SEE SECTION 5 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESSES

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

STARTING ADDRESSES

1. STARTING ADDRESS 200

A. SET SWITCHES (SEE SECTION 5)

B. PRESS START

C. THE PROGRAM WILL TYPE:

TEST ALL DRIVES? (Y OR N)

D. IF THE OPERATOR TYPES "Y" THE PROGRAM WILL TEST ALL RS03 DRIVES ON THE SYSTEM

E. IF THE OPERATOR TYPES "N" THE PROGRAM WILL TYPE

TYPE UNIT #

THE PROGRAM WILL ONLY TEST THAT DRIVE. THE PROGRAM WILL THEN TYPE:

```
                                        "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD
                                        BE ON - CHECK - THEN HIT CONT"
```

THE OPERATOR SHOULD CHECK THESE LIGHTS TO MAKE SURE THAT THEY ARE ALL ON - THEN HIT CONTINUE. THE PROGRAM WILL THEN START TESTING THE UNIT THAT WAS SELECTED.

2. STARTING ADDRESS 220

A. SET SWITCHES (SEE SECTION 5)

B. PRESS START

C. THE PROGRAM WILL THEN TEST ALL RS03 DRIVES ON THE SYSTEM.

5. OPERATIONAL SWITCH SETTINGS

SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... LOOP ON TEST
SW<13> = 1 ..... INHIBIT TYPEOUTS
SW<12> = 1 ..... TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
SW<11> = 1 ..... RUN MAINTENANCE MODE VERIFY TEST
SW<10> = 1 ..... BELL ON ERROR
         0 ..... BELL ON PASS COMPLETE
SW<09> = 1 ..... LOOP ON ERROR
SW<08> = 1 ..... LOOP ON TEST IN SW<7:0>
```

5.1 SUBROUTINE ABSTRACTS

THIS PROGRAM USES TRAP INSTRUCTIONS TO EXECUTE CLOCKING AND REGISTER CHECKING. THE TRAP INSTRUCTIONS THAT WE USED, ARE LISTED BELOW WITH A BREIF DESCRIPTION OF WHAT EACH ONE DOES.

5.1.1 CLRDK

TRAPS TO A TAG CALLED ".CLRDK". THIS ROUTINE CLEARS ALL REGISTERS BY SETTING THE "CLEAR BIT" IN RSCS2. (MOV#40,@RHCS2) THE NUMBER OF THE UNIT UNDER TEST IS THEN RELOADED INTO RSCS2 AND THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE CLRDK INSTRUCTION.

5.1.2 MRDMD

TRAPS TO A TAG CALLED ".MRDMD". THIS ROUTINE PUTS THE DRIVE INTO MAINTENANCE MODE BY LOADING #000001 INTO RSMR AND THEN RETURNS TO THE

```
272                     NEXT INSTRUCTION FOLLOWING THE MRDMD INSTRUCTION.
```

5.1.3 MRINT

TRAPS TO A TAG CALLED ".MRINT". CLOCKS THE MAINTENANCE REGISTER TWICE WITH AN 11 AND A 1 AND RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MRINT INSTRUCTION.

5.1.4 MRIND

TRAPS TO A TAG CALLED ".MRIND". CLOCKS AN INDEX PULSE INTO THE MAINTENANCE REGISTER THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MRIND INSTRUCTION.

5.1.5 MRCLK

TRAPS TO A TAG CALLED ".MRCLK". CLOCKS THE MAINTENANCE REGISTER WITH AN 11 AND A 1, UPDATES THE CLOCK COUNTER, AND THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MRCLK INSTRUCTION.

5.1.6 MRCK

TRAPS TO A TAG CALLED ".MRCK". THIS ROUTINE CHECKS THE MAINTENANCE REGISTER TO EQUAL THE VALUE FOLLOWING THE MRCK INSTRUCTION. IF THE MAINTENANCE REGISTER DOES NOT COMPARE, THE PROGRAM RETURNS TO THE "HLT" INSTRUCTION FOLLOWING THE CORRECT VALUE AND PRINTS OUT THE ERROR. IF THE MAINTENANCE REGISTER IS CORRECT, THE PROGRAM RETURNS TO THE INSTRUCTION FOLLOWING THE "HLT" INSTRUCTION.

5.1.7 DSCK

TRAPS TO A TAG CALLED ".DSCK". THIS ROUTINE CHECKS THE DRIVE STATUS REGISTER AND WORKS THE SAME WAY AS THE MRCK ROUTINE.

5.1.8 XBIT

TRAPS TO A TAG CALLED ".XBIT". THIS ROUTINE GETS ONE DATA BIT THAT IS CURRENTLY BEING WRITTEN FROM THE DATA BUFFER IN CORE AND STORES IT IN A LOCATION CALLED NOWOD. THE PREVIOUS CONTENTS OF NOWOD IS STORED IN LASTOD. THIS INFORMATION IS USED BY THE CLKD1 AND CLKD0 ROUTINES TO DETERMINE THE CORRECT STATE OF THE MWDB (BIT 12) BIT IN RSMR WHEN WRITING. THIS ROUTINE MAKES BITS 16 AND 17 OF EACH DATA WORD (RS03 WRITES 18 BIT WORDS) EQUAL ZERO. THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE XBIT INSTRUCTION.

5.1.9 CLKD1 AND CLKD0

TRAPS TO LOCATIONS ".CLKD1" AND ".CLKD0". THESE TWO ROUTINES USE THE DATA BITS RECEIVED FROM THE XBIT ROUTINE TO DETERMINE THE CORRECT STATE OF MWDB (BIT 12) IN RSMR WHEN WRITING. THESE ROUTINES ALSO CALCULATE THE CORRECT STATES OF THE CRCW, SB, AND LSR BITS IN RSMR AND DOES A COMPARE FOR THE CORRECT ANSWER. IF THE MAINTENANCE REGISTER DOES NOT COMPARE, THE PROGRAM RETURNS TO THE "HLT" INSTRUCTION FOLLOWING THE TRAP AND TYPES OUT THE ERROR. IF THE MAINTENANCE REGISTER WAS CORRECT, THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE "HLT."

5.1.10 RBIT

TRAPS TO A TAG CALLED ".RBIT". THIS ROUTINE GETS THE ONE DATA BIT THAT ARE CURRENTLY BEING "READ" FROM THE DISK FROM THE INBUF DATA TABLE IN CORE AND STORES THAT BIT IN A LOCATION CALLED NOWOD. THE PROGRAM THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE RBIT INSTRUCTION.

5.1.11 CLKR1 AND CLKR0

TRAPS TO LOCATIONS ".CLKR1" AND ".CLKR0". THESE TWO ROUTINES USING THE DATA BITS RECRIVED FROM THE RBIT ROUTINE SET AND CLEAR THE MRDB (BIT 2) BIT IN RSMR IN THE PROPER SEQUENCE CORRESPONDING TO THE DATA PATTERN WHICH IS BEING "READ". THESE ROUTINES ALSO CALCULATE THE CORRECT STATES OF THE CRCW AND SB BITS IN RSMR AND DOES A COMPARE FOR THE CORRECT ANSWER. IF THE MAINTENANCE REGISTER DOES NOT COMPARE, THE PROGRAM RETURNS TO THE "HLT" INSTRUCTION FOLLOWING THE TRAP AND TYPES OUT THE ERROR. IF THE MAINTENANCE REGISTER WAS CORRECT, THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE HLT.

5.1.12 MCLK1

TRAPS TO A TAG CALLED ".MCLK1". THIS ROUTINE CLOCKS THE MAINTENANCE REGISTER BY MOVING A 11 INTO RSMR. UPDATES THE CLOCK COUNTER AND THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MCLK1 INSTRUCTION.

5.1.13 MCLK0

TRAPS TO A TAG CALLED ".MCLK0". THIS ROUTINE CLOCKS THE MAINTENANCE REGISTER BY MOVING A 1 INTO RSMR. RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MCLK0 INSTRUCTION.

5.1.14 MCLKB

TRAPS TO A TAG CALLED ".MCLKB". CLOCKS THE MAINTENANCE REGISTER WITH A 1 AND A 11, UPDATES THE CLOCK COUNTER, AND THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MCLKB INSTRUCTION.

5.1.15 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.1.16 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.1.17 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

```
ADR   CS1 = ----- CS2 = ----- ER = -----
GOOD      = ----- BAD = -----
```

WHERE:

CS1,CS2,ER ETC.	= RH11/RS03 REGISTERS.
GOOD	= EXPECTED DATA.
BAD	= DATA RECEIVED.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200 OR AT 220

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN ONE AND A HALF MINUTES WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITALLY SET TO 500

9. TEST DESCRIPTION

1. TEST FOR ONLINE DRIVES

   SET ERROR BITS IN RSER. THIS CAUSES ATTENTION SUMMARY BITS TO SET IN RSAS. DO FOR ALL DRIVES. RSAS HAS NOT YET BEEN TESTED. SO IN THE CASE OF NO BITS IN RSAS SETTING, DRIVE 0 IS TESTED.

2. RESET TEST FOR REGISTERS

   SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. DO A RESET AND TEST ALL R/W BITS TO BE CLEARED.

3. SET AND CLEAR ALL REGISTERS

   SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB AND RSMR AND TEST. SET ALTERNATE BITS AND CHECK TO MAKE SURE BITS ARE NOT TIED TOGETHER. NOW SET ALL BITS AND CLEAR THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.

4. TEST "CLEAR BIT" IN RSCS2

```
496                     SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC
497                     RSDB, AND RSMR.  SET CLEAR BIT IN RSCS2.  NOW TEST ALL R/W
```

BITS FOR 0 IN ALL THE ABOVE REGISTERS.

5. LOAD RSDB WITH ALL ONES AND ALL ZEROS

LOAD RSDB WITH A WORD OF ZEROS AND A WORD OF ONES. WAIT FOR "OR" TO SET AND THEN CHECK OUTPUT OF SILO. IF OR DID NOT SET ERROR MESSAGE APPEARS.

6. TEST PROGRAM INTERRUPT

THE PROGRAM FORCES A INTERRUPT BY MOVING A 300 INTO RSCS1.

7. MAINTENANCE TIMING TEST

THE FOLLOWING TEST ON THE RS03 DISK IS A SINGLE-STEPPED MAINTENANCE MODE TEST ON THE RS03 TIMING LOGIC. THE ACTUAL DISK SURFACE IS SUBSTITUTED BY THE MAINTENANCE REGISTER, I.E., THE PROGRAM WILL SUPPLY ALL "DISK CLOCK" PULSES TO DRIVE THE TIMING LOGIC. WE ARE TESTING THE ENTIRE "TIMING TRACK", INDEX PULSE FUNCTION, RESYNC AREA, SECTOR COUNTER, ETC.

-PUT DRIVE INTO MAINTENANCE MODE.
-ASSERT INDEX PULSE TO INITIALIZE DRIVE TIMING LOGIC.
-INDEX PULSE SHOULD CLEAR LOOK-AHEAD REGISTER.
-CLOCK TIMING TO STEP THROUGH RESYNC PERIOD.
-CHECK FOR SECTOR PULSE.
-PERFORM MAINTENANCE CLOCK OPERATION TO CHECK FOR 64 SECTOR PULSES.
-THE LOOK-AHEAD REGISTER SHOULD NOW POINT TO THE CURRENT SECTOR.
-REPEAT STEPS TO CLOCK THROUGH ALL THE SECTORS TO CHECK SECTOR COUNT.

8. SECTOR FRACTION TEST

CLOCK THROUGH AN ENTIRE TRACK IN MAINTENANCE MODE WHILE CHECKING FOR THE PROPER OPERATION OF THE LOOK-AHEAD REGISTER AND THE SECTOR FRACTION COUNTER.

-INITIALIZE DRIVE AND STEP THROUGH RESYNC AREA.
-CHECK FOR SECTOR PULSE.
-LOOK-AHEAD REGISTER SHOULD = 0.
-STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA AREA WHILE CHECKING THE SECTOR FRACTION.
-CHECK FRACTIONS TO CHANGE AFTER THE CORRECT NUMBER OF MAINTENANCE CLOCKS.

WHEN THE LAST WORD IS BEING TRANSFERRED, SECTOR AND FRACTION IS EQUAL TO 7777 TO INDICATE LAST WORD ON THIS TRACK -- HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER WILL CLEAR THE FRACTION BITS IF ANOTHER WORD IS CLOCKED. RSLA

554 SHOULD INDICATE 7700 ON ANOTHER MAINTENANCE CLOCK.

DESCRIPTION

9. DISK ILLEGAL FUNCTION TEST

TEST ILLEGAL FUNCTION (ILF) IN RSER. SEND AN ILLEGAL FUNCTION CODE TO THE DRIVE CONTROL REGISTER WITHOUT SETTING THE GO BIT. THE "ILF" BIT SHOULD NOT BE SET. THE "GO" BIT IS THEN SET. A CHECK IS THEN MADE FOR "ATA" AND "ERR" TO BE SET IN THE DRIVE STATUS REGISTER (RSDS) AND "ILF" IN THE DRIVE ERROR REGISTER (RSER). ALL ILLEGAL FUNCTION CODES ARE CHECKED.

10. TEST THE DRIVE NO-OP CODES 1 AND 21

THIS IS TESTED WITH AND WITHOUT ERRORS BEING SET TO PROVE THAT IT DOESN'T CHANGE ANYTHING.

11. DRIVE SEARCH TEST 1

A DRIVE SEARCH FUNCTION IS GIVEN TO THE DRIVE FOR SECTOR 3. (SECTOR 41, IF SECTOR INTERLEAVING IS ENABLED) THE POSITIONING IN PROGRESS BIT (PIP) AND THE DRIVE READY BIT (DRY) IN THE DRIVE STATUS REGISTER (RSDS) ARE CHECKED. THE ADDRESS CONFIRM BIT (AC) IS ALSO CHECKED.

12. DRIVE SEARCH TEST 2

THIS TEST INITIALIZES A DRIVE SEARCH FUNCTION FOR SECTOR 0 WHEN THE DRIVE IS CURRENTLY AT THE DESIRED SECTOR, THE SEARCH FUNCTION SHOULD NOT BE COMPLETED UNTIL THE DRIVE MAKES A COMPLETE REVOLUTION AND REACHES THE BEGINNING OF THE DESIRED SECTOR.

13. REGISTER MODIFICATION REFUSED TEST

RMR IN THE DRIVE ERROR REGISTER (RSER) SHOULD SET BY TRYING TO MODIFY ONE OF THREE DRIVE REGISTERS WHILE THE DRIVE IS BUSY DURING A DRIVE SEARCH FUNCTION.

1. RSCS1
2. RSDA
3. RSER

TEST THAT RMR DOES NOT SET WHEN MODIFYING THE ATTENTION SUMMARY REGISTER (RSAS).

14. DRIVE SELECT TEST

THE PROGRAM LOADS A DRIVE REGISTER, OF THE DRIVE UNDER TEST, TO ALL ONES. THE PROGRAM THEN FINDS A NON-EXISTENT DRIVE AND TRIES TO LOAD ITS REGISTER WITH ALTERNATE ONES AND ZEROS. THIS SHOULD CAUSE "NED" TO SET IN RSCS2. THE PROGRAM RE-SELECTS THE DRIVE UNDER TEST AND CHECKS ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.

611
612

15. MAINTENANCE WRITE TEST

DESCRIPTION

THIS IS AN RS03 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR WRITE TEST. WE ARE TESTING THE COMPLETE DATA PATH FOR A DATA TRANSFER TO THE DISK. MILLER ENCODED DATA TO BOTH SURFACES IS CHECKED ALONG WITH CORRECT GENERATION OF THE CRC WORD AT THE END OF THE SECTOR. INDEX PULSES, RESYNC, TIMING PREAMBLE, AND SECTOR PULSES ARE ALSO CHECKED.

16. MAINTENANCE READ TEST

THIS IS AN RS03 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR READ TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE DISK DECODING LOGIC TO CORE MEMORY. (THE PHASE LOCK LOOP IS NOT TESTED IN MAINTENANCE MODE.)

17. MAINTENANCE MODE DATA WRITE CHECK TEST

A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION. WITHIN THE RS03, A WRITE CHECK FUNCTION IS IDENTICAL TO A READ FUNCTION.

18. MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)

THE RS03 DISK IS SET UP TO READ (IN MAINTENANCE MODE) ONE SECTOR OF A SPECIALLY CREATED DATA PATTERN WHICH LEAVES ONLY ONE BIT SET IN THE CRC REGISTER PRIOR TO CHECKING THE CRC WORD. THE CORRESPONDING CRC WORD IS THEN "READ", RESULTING IN NO DCK ERROR. THE DATA PATTERN IS THEN MODIFIED (BY SHIFTING) AND THE ENTIRE READ SEQUENCE REPEATED UNTIL ALL 16 BITS IN THE CRC REGISTER HAVE BEEN CHECKED.

19. MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)

THIS TEST IS SIMILAR TO CRC TEST 1 EXCEPT THAT THE DATA PATTERN HAS BEEN MODIFIED TO LEAVE A SINGLE BIT SET IN THE CRC REGISTER AFTER BOTH DATA AND CRC WORDS HAVE BEEN "READ". THIS CAUSES A DCK ERROR. THE READ SEQUENCE IS REPEATED 16 TIMES TO TEST THAT EACH BIT IN THE CRC REGISTER CAN CAUSE A DCK ERROR.

20. IGNORE FUNCTION TEST

PUT THE DISK IN MAINTENANCE MODE AND SET ERROR CONDITIONS IN THE DRIVE ERROR REGISTER (RSER). TRY TO START A READ TRANSFER. THE "GO" BIT IN RSCS1 SHOULD NOT SET. MISSED TRANSFER ERROR (MXF) SHOULD SET IN RSCS2 WHICH IN TURN SHOULD CAUSE "TRE" AND "SC" TO SET IN RSCS1.

21. INVALID ADDRESS TEST

FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK ADDRESS REGISTER (RSDA). THIS SHOULD CAUSE "IAE" TO SET IN THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO

```
                        RSCS1  WHICH  IN  TURN  SHOULD  CAUSE ATTENTION TO SET IN THE
                        DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE
```

CONTROL REGISTER (RSCS1).

22. DISK OPERATION INCOMPLETE (OPI) ERROR TEST

PUT DISK IN MAINTENANCE MODE AND START A READ COMMMAND. THEN ISSUE THREE DISK "INDEX" PULSES TO SIMULATE A COMPLETE ROTATION OF THIS DISK SURFACE. THE THIRD INDEX PULSE SHOULD CAUSE OPERATION INCOMPLETE (OPI) TO SET IN THE DRIVE ERROR REGISTER (RSER) AND "ATA" AND "ERR" IN THE DRIVE STATUS REGISTER (RSDS).

23. PARITY ERROR TEST

SET "PAT" BIT IN RSCS2. WRITE A DRIVE REGISTER. "PAR" SHOULD SET IN THE DRIVE ERROR REGISTER (RSER) WHICH SHOULD CAUSE "ATA" TO SET IN RSAS AND 'SC' TO SET IN RSCS1.

24. MAINTENANCE MODE INTERRUPT TEST

IN THIS TEST THE INTERRUPT ENABLE (I.E.,) BIT IS SET. A TWO SECTOR WRITE COMMAND IS GIVEN. AN "RMR" ERROR IS THEN CAUSED WHILE THE FIRST SECTOR IS BEING WRITTEN. WHEN THE FUNCTION IS COMPLETED, THE DRIVE SHOULD INTERRUPT.

25. DISK ADDRESS OVERFLOW (AOE) TEST

SET UP TO TRANSFER 2 SECTORS TO THE DISK, STARTING AT TRACK 77 SECTOR 77 TO CAUSE A DISK ADDRESS OVERFLOW CONDITION. ALSO CHECK LAST BLOCK TRANSFER (LBT) BIT TO SET IN THE RSDS REGISTER.

26. MAINTENANCE VERIFY TEST

THIS TEST WILL ONLY RUN IF SWITCH 11 IS SET IN THE "SWITCH REGISTER" FOR IT WILL ACTUALLY WRITE DATA ONTO THE DISK. IT WILL WRITE ONE TRACK OF ALL ONES. THE DRIVE IS THEN PLACED IN MAINTENANCE MODE AND IT WILL THEN WRITE ONE SECTOR OF THE SAME TRACK WITH ALL ZEROS. THE DRIVE IS THEN TAKEN OUT OF "MAINTENANCE MODE" AND THE TRACK IS THEN READ. THE TRACK SHOULD CONTAIN ALL ONES.

```
                        %
                .TITLE  MAINDEC-11-DERSC-B      RS11-RS03 MAINTENANCE MODE DIAGNOSTIC
                ;COPYRIGHT 1974,1975,1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                ;PROGRAM BY STANLEY HARACKIEWICZ


                ;          SWITCH                       USE
                ;          ------                       ---------------------
                        SW15=   100000          ;HALT ON ERROR
                        SW14=   40000           ;LOOP ON TEST
100000
040000
```

```
                020000                          SW13=   20000           ;INHIBIT ERROR TYPEOUTS
                010000                          SW12=   10000           ;TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
                004000                          SW11=   4000            ;RUN MAINTENANCE MODE VERIFY TEST
                002000                          SW10=   2000            ;0 - BELL ON PASS COMPLETE
                                                                        ;1 - BELL ON ERROR
                001000                          SW9=    1000            ;LOOP ON ERROR
                000400                          SW8=    400             ;LOOP ON TEST IN SW<7:0>
                000000                  .=      0                       ;TRAP CATCHER FROM 0 - 776
                000200                  .=      200
000200  000137  000232                          JMP     @#BEGIN1

                000220                  .=      220
000220  052767  000100  000720                  BIS     #BIT6,FLAG3     ;TEST ALL DRIVES
000226  000137  001236          BEGIN2: JMP     @#BEGIN

000232  042767  000100  000706  BEGIN1: BIC     #BIT6,FLAG3     ;CLEAR MULTI DRIVE FLAG
000240  000772                          BR      BEGIN2
```

```
                                
000001                          N=      1               ;INITALIZE FOR NEWTST
104000                          HLT=    EMT             ;SET HLT TO EMT FOR ERROR TYPEOUTS
177776                          PS=     177776          ;PROCESSOR STATUS
177776                          PSW=    PS              ;PROCESSOR STATUS WORD
177570                          SWR=    177570          ;SWITCH REGISTER
177570                          DISPLAY=SWR             ;DISPLAY REGISTER
000007                          BELL=   7               ;BELL
000000                          R0=     %0              ;R0 - DEFINE REGISTERS
000001                          R1=     %1              ;R1
000002                          R2=     %2              ;R2
000003                          R3=     %3              ;R3
000004                          R4=     %4              ;R4
000005                          R5=     %5              ;R5
000006                          SP=     %6              ;R6 - STACK POINTER
000007                          PC=     %7              ;R7 - PROGRAM COUNTER
000001                          BIT0=   1               ;BIT EQUATES
000002                          BIT1=   2
000004                          BIT2=   4
000010                          BIT3=   10
000020                          BIT4=   20
000040                          BIT5=   40
000100                          BIT6=   100
000200                          BIT7=   200
000400                          BIT8=   400
001000                          BIT9=   1000
002000                          BIT10=  2000
004000                          BIT11=  4000
010000                          BIT12=  10000
020000                          BIT13=  20000
040000                          BIT14=  40000
100000                          BIT15=  100000
000001                  GOOD=   %1              ;FOR GOOD DATA
000000                  BAD=    %0              ;FOR BAD DATA
                                
```

```
                001000                  .=      1000
                                        ICNT:   0               ;LH = ITERATION COUNT  ;RH = TEST NO.
001000  000000                          ERRORS: 0               ;ERROR COUNT
001002  000000                          PCNT:   0,0             ;2 WORD PASS COUNT
001004  000000  000000                  LAD:    0               ;LOOP ADDRESS FOR SCOPE
001010  000000                          HLTADR: 0               ;ADDRESS OF LAST HLT INSTRUCTION  EXECUTED
001012  000000                          FILCHR: 1000            ;FILCHR=0 (CHAR)  ;FILCHR+1=2 (COUNT)
001014  001000                          TPS:    177564          ;OUTPUT STATUS REGISTER
001016  177564                          TPB:    177566          ;OUTPUT BUFFER
001020  177566
                001100                  .=      1100
                                        ;DISK I/O REGISTERS
001100  172040                          RSCS1:  172040          ;DISK CONTROL + STATUS REGISTER
001102  172050                          RSCS2:  172050          ;DISK CONTROL + STATUS REGISTER
001104  172042                          RSWC:   172042          ;WORD COUNT REGISTER
001106  172044                          RSBA:   172044          ;BUS ADDRESS
001110  172046                          RSDA:   172046          ;DISK ADDRESS (DESIRED ADDRESS)
001112  172052                          RSDS:   172052          ;DRIVE STATUS
001114  172054                          RSER:   172054          ;ERROR REG.
001116  172056                          RSAS:   172056          ;ATTENTION SUMMARY
001120  172060                          RSLA:   172060          ;LOOK AHEAD
001122  172062                          RSDB:   172062          ;DATA BUFFER REGISTER
001124  172064                          RSMR:   172064          ;MAINTENANCE REGISTER
001126  172066                          RSDT:   172066          ;DRIVE TYPE REGISTER
001130  000204                          RSVEC:  204             ;INTERRUPT VECTOR
001132  000206                          RSVCPS: 206             ;INTERRUPT PRIO. VECTOR
001134  172041                          RSCS1B: 172041          ;ODD BYTE ADD FOR CS1
001136  172051                          RSCS2B: 172051          ;ODD BYTE ADD FOR CS2
001140  172043                          RSWCB:  172043          ;ODD BYTE ADD FOR CW
001142  172045                          RSBAB:  172045          ;ODD BYTE ADD FOR BA
```

```
                                ;BIT ASSIGNMENTS FOR ERROR TYPEOUTS
                                ;THE RS REGISTERS ARE DIVIDED INTO 3 GROUPS.
                                ;CS1,CS2 AND ER ARE IN THE FIRST GROUP.THIS GROUP IS ALWAYS
                                ;TYPED WITH EITHER OF THE OTHER GROUPS. AS,BA,DA, WC AND DS
                                ;ARE IN THE SECOND GROUP. DT,DB,MR, AND LA ARE IN THE 3RD
                                ;GROUP.YOU CAN NOT INTERMIX GROUP 2 OR 3. THEY HAVE
                                ;TO BE TYPED SEPERATELY.
                                ;EXAMPLE:   HLT !CS1,AS,BA
                                ;           HLT !CS1!DT!DB

000001                              CS1=1           ;CONTROL AND STATUS 1
000002                              ER=2            ;CONTROL AND STATUS 2
000004                              DA=4            ;DESIRED ADD
000010                              WC=10           ;WORD COUNT
000020                              BA=20           ;BUS ADDRESS
000040                              DS=40           ;DRIVE STATUS
000100                              AS=100          ;ATTENTION SUMMARY
000200                              CS2=200         ;CONTROL AND STATUS REG
000204                              LA=204          ;LOOK AHEAD
000210                              DB=210          ;DATA BUFFER
000220                              MR=220          ;MAINTENANCE
000240                              DT=240          ;DRIVE TYPE

                                ;BIT ASSIGNMENTS FOR THE REGISTER BITS

040000                              TRE=40000       ;TRANSFER ERROR CS1
100000                              SC=100000       ;SPECIAL CONDITIONS CS1
000100                              IR=100          ;INPUT READY CS2
000200                              OR=200          ;OUTPUT READY CS2
002000                              PGE=2000        ;PROGRAM ERROR-CS2
010000                              NED=10000       ;NON-EXISTENT DRIVE CS2
040000                              WCE=40000       ;WRITE CHECK ERROR-CS2
100000                              DLT=100000      ;DATA LATE ERROR CS2
000200                              DRY=200         ;DRIVE READY DS
020000                              PIP=20000       ;POSITIONING IN PROGRESS DS
002000                              LBT=2000        ;LAST BLOCK TRANSFER-DS
040000                              ERR=40000       ;ERROR DS
100000                              ATA=100000      ;ATTENTION ACTIVE-DS
001000                              DAO=1000        ;DISK OVERFLOW ERROR-ER
100000                              DCK=100000      ;DATA CHECK ERROR-ER
000010                              BAI=10          ;BUS ADDR INCREMENT INHIBIT
000100                              IE=100          ;INTERRUPT INABLE CS1
```

```
                                        ;WORKING LOCATIONS
                                        
001144  000000                FLAG2:  0           ;SECOND FLAG WORD
001146  000000                FLAG3:  0           ;3RD FLAG WD
001150  000000                LSTEV:  0           ;LAST EVEN BIT TRANSFERED
001152  000000                LSTOD:  0           ;LAST ODD BIT TRANSFERED
001154  000000                NOWEV:  0           ;PRESENT EVEN BIT BEING XFERED
001156  000000                NOWOD:  0           ;PRESENT ODD BIT BEING XFERED
001160  000000                RSO:    0           ;SAME
001162  000000                UNNUM:  0           ;UNIT CURRENTLY BEING TESTED
001164  000000                UNITSV: 0           ;SET BIT=UNIT ON BUS
001166  000000                UNCMP:  0           ;FOR COMPARING FOR # OF DEVICE
001170  000000                ONCEE:  0           ;DID WE TEST ANY DRIVES
001172  000000                TIMSV:  0           ;SAVE LOC FOR TIME
        172100                MPRD=172100         ;PARITY REG
001174  000000                SAVEE:  0           ;WORK LOC
001176  000000  000000        MCCNT:  0,0         ;MAINT CLOCK COUNT
001202  000000                WCRC:   0           ;WORK LOC FOR CREATING CRC WORD
001204  000000                REPT:   0           ;REPEAT COUNTER
001206  000000                REPT1:  0           ;REPEAT COUNTER
001210  000000                CLKCNT: 0           ;CLOCK COUNTER FOR EACH WORD
001212  000000                INBIT:  0           ;USED IN CRC CAL ROUTINE
001214  000000                WK15:   0           ;USED IN CRC CAL ROUTINE
001216  000000                WORK:   0
001220  000000                WORK0:  0
001222  000000                WORK1:  0
001224  000000                WORK2:  0
001226  000000                WORK3:  0
001230  000000                WORK4:  0
001232  000000                WORK5:  0
001234  000000                WORK6:  0
```

```
                        ;DISCRIPTION OF BITS IN LOCATION ONCEE

                        ;BIT0 MEANS FOUND DRIVE
                        ;BIT1 ERROR DO NOT CHANGE ILLEGAL FUNCTION
                        ;BIT2 ERROR FLAG
                        ;BIT3 TESTING CODE 21 FLAG
                        ;BIT5 TYPEOUT CLOCK COUNT
                        ;BIT6 1ST TRANSFER WORD FLAG
                        ;BIT7 WRITTING LAST WORD OF SECOTR
                        ;BIT8 TRANSFERRING CRC WORD
                        ;BIT9 FOR INTERLEAVED DRIVES
                        ;BIT10 1ST TIME FLAG IN SECTOR FRACTION TEST
                        ;BIT11 DO TKSEL TEST
                        ;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
                        ;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
                        ;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
                        ;BIT15 MEANS ERROR FOUND

                        ;DISCRIPTION OF BITS IN LOCATION FLAG2

                        ;BIT0 SWITCH FOR RWCLK IN MR REG
                        ;BIT1 MAINTENANCE MODE VERIFY TEST
                        ;BIT2 IN WRITE CK TEST FOR CLKR1 ROUTINE
                        ;BIT3 DONE 1ST CRC WD IN CRC TEST
                        ;BIT4 1ST TIME THROUGH IN CRC TEST
                        ;BIT5 [illegible] CRC TEST
                        ;BIT7 DOING FIRST XFER WD IN XBIT
                        ;BIT8 XFER DATA BITS 16 AND 17 IN XBIT ROUTINE
                        ;BIT9 SAME
                        ;BIT10 XFER CRC BITS 16 AND 17 IN XBIT ROUTINE
                        ;BIT11 USED IN RBIT ROUTINE FOR DATA BITS 17 AND 16
```

```
 919  001236  012706  000500          BEGIN:  MOV     #500,SP         ;SET STACK TO  *** 500 ***
 920  001242  012737  025066  000024          MOV     #.POWER,@#24    ;SET UP PF VECTOR
 921  001250  012737  000340  000026          MOV     #340,@#26       ;LOCK OUT THE WORLD
 922  001256  012737  024516  000030          MOV     #.HLT,@#30      ;SET EMT VECTOR
 923  001264  012737  000340  000032          MOV     #340,@#32       ;LOCK UP
 924  001272  012737  025470  000034          MOV     #.TRAP,@#34     ;SET TRAP VECTOR
 925  001300  012737  000340  000036          MOV     #340,@#36       ;LOCK UP
 926  001306  005067  177466                  CLR     ICNT            ;INIT ICNT
 927  001312  005067  177472                  CLR     LAD             ;INIT LAD
 928  001316  042767  000020  177622          BIC     #BIT4,FLAG3     ;CLEAR TEST ONLY ONE DRIVE FLAG
 929  001324  042767  177677  177612          BIC     #177677,FLAG2
 930  001332  042767  153777  177630          BIC     #153777,ONCEE
 931  001340  032767  000100  177600          BIT     #BIT6,FLAG3     ;TEST ALL DRIVES?
 932  001346  001402                          BEQ     5$              ;ASK
 933  001350  000137  001702                  JMP     @#MULTII
 934  001354                          5$:
 935  001354  104402  001360                  TYPE    ,.+2            ;.ASCIZ <15><12>"TEST ALL DRIVES? (Y OR N) "
 936  001416  104412                          RDLIN
 937  001420  122767  000131  024022          CMPB    #'Y,INPUT       ;TEST FOR YES
 938  001426  001525                          BEQ     MULTII          ;YES
 939  001430  052767  000020  177510          BIS     #BIT4,FLAG3     ;SET TEST ONLY ONE DRIVE FLAG
 940  001436                          1$:
 941  001436  104402  001442                  TYPE    ,.+2            ;.ASCIZ "TYPE UNIT #"
 942  001456  104410                          RDOCT
 943  001460  012604                          MOV     (6)+,R4         ;GET NUMBER
 944  001462  022704  000010                  CMP     #10,R4          ;CORRECT #
 945  001466  101763                          BLOS    1$              ;NO
 946  001470  010467  177466                  MOV     R4,UNNUM        ;SET UNIT #
 947  001474  005002                          CLR     R2              ;CLEAR WORK AREA
 948  001476  000261                          SEC                     ;SET CARRY
 949  001500  006102                  2$:     ROL     R2              ;SET WORK BIT
 950  001502  005704                          TST     R4              ;IS THIS BIT CORRESPOND WITH CORRECT DRIVE #
 951  001504  001402                          BEQ     3$              ;YES
 952  001506  005304                          DEC     R4              ;NO TRY AGAIN
 953  001510  000773                          BR      2$              ;TEST AGAIN
 954  001512  010267  177446          3$:     MOV     R2,UNITSV       ;SET DRIVE BIT IN UNITSV
 955  001516  010267  177444                  MOV     R2,UNCMP        ;SET UNIT COMPARE
 956  001522  016777  177434  177352          MOV     UNNUM,@RSCS2    ;LOAD DRIVE
 957  001530  012777  177777  177356          MOV     #-1,@RSER       ;LOAD ERRORS
 958  001536  104402  001542                  TYPE    ,.+2            ;.ASCIZ "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD BE ON
 959  001662  000000                          HALT                    ;WAIT FOR LIGHTS TO BE CHECKED
 960  001664  026777  177304  177234          CMP     UNITSV,@RSAS    ;DID CORRECT ATA SET
 961  001662  001405                          BEQ     4$
 962  001664  017700  177226                  MOV     @RSAS,BAD       ;GET RSAS
 963  001670  016701  177270                  MOV     UNITSV,GOOD     ;GET CORRECT AND
 964  001674  104000                          HLT                     ;RSAS=BAD GOOD=CORRECTIONS
 965                                                                  ;ATA BIT SHOULD SET FOR ERRORS
 966                                                                  ;WERE SET IN RSER
 967  001676  000167  000430          4$:     JMP     NOWGO           ;START TESTING
```

```
                                        ;NOW TEST FOR DRIVES
                                
001702  012701  000010          MULTII: MOV     #8.,R1          ;PUT 8 INTO R1 FOR COUNT
001706  005077  177170                  CLR     @RSCS2          ;SET DEVICE TO ZERO
001712  012777  177777  177174  TRY:    MOV     #-1,@RSER       ;CAUSE AN ERROR +SETS BIT IN RSAS REG
001720  005301                          DEC     R1              ;DO A MAXIMUM OF 8 TIMES
001722  001403                          BEQ     DVNUM           ;TESTED FOR ALL DRIVES GET OUT
001724  005277  177152                  INC     @RSCS2          ;INCREMENT DRIVE UNIT
001730  000770                          BR      TRY             ;REPEAT FOR NEXT DRIVE
001732  017767  177160  177224  DVNUM:  MOV     @RSAS,UNITSV    ;SAVE
001740  012767  000401  177220          MOV     #401,UNCMP      ;SETUP TO CMP WITH UNITSV
001746  012767  000000  177206          MOV     #0,UNNUM        ;PUT 0 INTO UNIT NO.
001754  032767  020000  175606          BIT     #BIT13,SWR      ;INHIBIT TYPE OUT?
001762  001015                          BNE     STTEST          ;YES
001764  104402  001770                  TYPE    ,.+2            ;.ASCIZ <15><12>"TESTING UNIT "
002010  042767  100000  177152          BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
002016  036767  177144  177140  STTEST: BIT     UNCMP,UNITSV    ;IS THIS DRIVE ON THE SYSTEM
002024  001440                          BEQ     TRYNX           ;NO
002026  016777  177130  177046          MOV     UNNUM,@RSCS2    ;YES PUT UNIT # INTO CS2
002034  022777  000000  177064  3$:     CMP     #0,@RSDT        ;IS THIS A RS03?
002042  001404                          BEQ     1$              ;YES
002044  022777  000001  177054          CMP     #1,@RSDT        ;IS IT A RS03?
002052  001025                          BNE     TRYNX           ;GET A NEW NUMBER
002054  032767  020000  175506  1$:     BIT     #BIT13,SWR      ;INHIBIT TYPE OUT?
002062  001020                          BNE     4$              ;YES
002064  032767  100000  177076          BIT     #BIT15,ONCEE    ;ANY ERRORS?
002072  001404                          BEQ     5$              ;NO
002074  104402  002100                  TYPE    ,.+2            ;.ASCIZ <15><12><12>
002104                          5$:
002104  016746  177052                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
002110  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
002112  104402  000040                  TYPE    ,40             ;TYPE SPACE
002116  042767  100000  177044          BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
002124  000502                  4$:     BR      NOWGO           ;NOW TEST
002126  032767  000020  177012  TRYNX:  BIT     #BIT4,FLAG3     ;MULTI DRIVE
002134  001074                          BNE     DONEE           ;NO
002136  006367  177024          1$:     ASL     UNCMP           ;CHECK NEXT BIT FOR DRIVE
002142  103403                          BCS     CHCKDV          ;DID WE TEST ANY REG?
002144  005267  177012                  INC     UNNUM           ;INC UNIT #
002150  000722                          BR      STTEST          ;CHECK FOR NEXT DRIVE
```

```
1008  002152  032767  000001  177010  CHCKDV: BIT     #BIT0,ONCEE     ;DID WE TEST ANY DRIVES?
1009  002160  001062                          BNE     DONEE           ;YES WE DID TEST A DRIVE
1010  002162  012767  100000  176776          MOV     #100000,UNCMP   ;NO DRIVES TESTED, COULD NOT SET
1011  002170  005067  176766                  CLR     UNNUM           ;ANY AS BITS, THUS DEFAULTS TO
1012  002174  032767  020000  175366          BIT     #BIT13,SWR      ;INHIBIT TYPE OUT?
1013  002202  001050                          BNE     4$              ;YES
1014  002204  016746  176752                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
1015  002210  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
1016  002212  104402  000040                  TYPE    ,40             ;TYPE SPACE
1017  002216  104402  002222                  TYPE    .+2             ;.ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
1018  002314  012767  000001  176644          MOV     #1,UNCMP        ;SETUP TO TEST UNIT 0
1019  002322  000000                          HALT                    ;WAIT
1020  002324  000402              4$:         BR      NOWGO           ;TEST DRIVE 0
1021  002326  000167  016454      DONEE:      JMP     DONE            ;GET OUT
1022
1023                              ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
1024                              ;TO CLEAR ALL THE RH AND RS REGISTERS
1025
1026  002332  052767  000001  176630  NOWGO:  BIS     #BIT0,ONCEE     ;SET FOUND DRIVE FLAG
1027  002340  016767  022150  176624          MOV     TIMES,TIMSV     ;SAVE TIME
1028  002346  012767  000001  022140          MOV     #1,TIMES        ;ONLY TEST ONCE
1029                              ;*************************************************************************
1030                              ;TEST 1         RESET TEST FOR REGISTERS
1031                              ;*************************************************************************
1032  002354  104400              TST1:   SCOPE
1033  002356  012737  000340  177776          MOV     #340,@#PS       ;LOCK OUT INTERUPTS
1034  002364  016777  176572  176510          MOV     UNNUM,@RSCS2    ;LOAD UNIT NO.
1035  002372  012777  177776  176500          MOV     #177776,@RSCS1  ;SET ALL
1036  002400  012777  177777  176500          MOV     #177777,@RSBA   ;POSSIBLE R/W
1037  002406  012777  177777  176474          MOV     #177777,@RSDA   ;BITS IN THESE REGISTERS
1038  002414  012777  177777  176472          MOV     #177777,@RSER
1039  002422  012777  177777  176474          MOV     #177777,@RSMR
1040  002430  012777  177777  176446          MOV     #177777,@RSWC
1041  002436  012777  177737  176436          MOV     #177737,@RSCS2
1042  002444  000005                          RESET                   ;CLEAR ALL BITS IN ALL REG.
1043
1044                              ;TEST RSCS2 FOR CLEARED BITS
1045
1046  002446  022777  000100  176426          CMP     #100,@RSCS2     ;DID THESE BITS GET CLEARED?
1047  002454  001401                          BEQ     .+4             ;YES
1048  002456  104200                          HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
1049  002460  016777  176476  176414          MOV     UNNUM,@RSCS2    ;PUT # OF UNIT IN TEST IN CS2
1050  002466  022777  010600  176416          CMP     #10600,@RSDS    ;IS DPR AND MOL SET?
1051  002474  001401                          BEQ     .+4             ;YES
1052  002476  104040                          HLT     !DS             ;NO WHY NOT?
1053
1054                              ;TEST CONTROL AND STATUS REG 1
1055  002500  022777  004200  176372          CMP     #4200,@RSCS1    ;DID THE READY BIT SET?
1056  002506  001401                          BEQ     .+4             ;YES
1057  002510  104001                          HLT     !CS1            ;READY SHOULD BE SET
```

```
                                        ;TEST BUS ADDRESS REGISTER
                                        ;
002512  005777  176370                  TST     @RSBA           ;IS BA REG. CLEARED
002516  001401                          BEQ     .+4             ;YES
002520  104020                          HLT     !BA             ;SHOULD BE 0
                                        ;
                                        ;TEST DISK ADDRESS REGISTER
                                        ;
002522  005777  176362                  TST     @RSDA           ;IS DA CLEARED
002526  001401                          BEQ     .+4             ;YES
002530  104004                          HLT     !DA             ;SHOULD BE 0
                                        ;
                                        ;TEST ERROR REG RSER
                                        ;
002532  005777  176356                  TST     @RSER           ;DID RSER CLEAR?
002536  001401                          BEQ     .+4             ;YES
002540  104002                          HLT     !ER             ;BITS(157015) SHOULD BE CLEARED
                                        ;
                                        ;TEST RS MAINTENANCE REGISTER
                                        ;
002542  032777  000077  176354          BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
002550  001401                          BEQ     .+4             ;YES
002552  104220                          HLT     !MR             ;BITS(77) SHOULD BE 0
                                        ;
                                        ;TEST WC REG IT SHOULD NOT CHANGE
                                        ;
002554  022777  177777  176322          CMP     #177777,@RSWC   ;DID IT CHANGE?
002562  001401                          BEQ     .+4             ;NO
002564  104010                          HLT     !WC             ;RESET SHOULD NOT MODIFY RSWC
                                        ;
                                        ;TEST RSAS
                                        ;
002566  005777  176324                  TST     @RSAS           ;IS REG CLEAR
002572  001401                          BEQ     .+4             ;YES
002574  104100                          HLT     !AS             ;NO
```

```
                                        ;*****************************************************************************
                                        ;TEST 2          TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
                                        ;*****************************************************************************
002576  104400                          TST2:   SCOPE

002600  012737  000340  177776  TTAGG:  MOV     #340,@#PS       ;LOCK OUT INTERRUPTS
002606  016777  176350  176266          MOV     UNNUM,@RSCS2    ;LOAD UNIT NO.
002614  012777  043576  176256          MOV     #43576,@RSCS1   ;SET ALL
002622  012777  020417  176252          MOV     #20417,@RSCS2   ;ALL
002630  012777  177777  176250          MOV     #177777,@RSBA   ;POSSIBLE
002636  012777  177777  176244          MOV     #177777,@RSDA   ;REGISTERS
002644  012777  177017  176242          MOV     #177017,@RSER
002652  012777  177777  176240          MOV     #177777,@RSDB
002660  012777  177777  176216          MOV     #177777,@RSWC
002666  012777  020417  176206          MOV     #20417,@RSCS2
002674  012777  000071  176222          MOV     #71,@RSMR
002702  012777  000040  176172          MOV     #40,@RSCS2      ;CLEAR ALL BITS
002710  022777  000100  176164          CMP     #100,@RSCS2     ;DID THE RIGHT BITS CLEAR?
002716  001401                          BEQ     .+4             ;YES
002720  104200                          HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
002722  016777  176234  176152          MOV     UNNUM,@RSCS2    ;GET DRIVE NUMBER
002730  032777  173577  176142          BIT     #173577,@RSCS1  ;DID ALL BITS GET CLEARED
002736  001401                          BEQ     .+4             ;YES
002740  104001                          HLT     !CS1            ;NO, ALL BITS SHOULD BE 0
                                        ;TEST BUS ADDRESS REGISTER

002742  005777  176140                  TST     @RSBA           ;IS BA REG. CLEARED
002746  001401                          BEQ     .+4             ;YES
002750  104020                          HLT     !BA             ;SHOULD BE 0

                                        ;TEST DISK ADDRESS REGISTER

002752  005777  176132                  TST     @RSDA           ;IS DA CLEARED
002756  001401                          BEQ     .+4             ;YES
002760  104020                          HLT     !BA             ;SHOULD BE 0

                                        ;TEST ERROR REG RSER

002762  032777  177777  176124          BIT     #177777,@RSER   ;DID THESE BITS GET CLEARED
002770  001401                          BEQ     .+4             ;YES
002772  104002                          HLT     !ER             ;BITS(157015) SHOULD BE CLEARED

                                        ;TEST RS MAINTENANCE REGISTER
002774  032777  000077  176122          BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
003002  001401                          BEQ     .+4             ;YES
003004  104220                          HLT     !MR             ;BITS(77) SHOULD BE 0

                                        ;TEST WC REG. IT SHOULD NOT CHANGE
003006  022777  177777  176070          CMP     #177777,@RSWC   ;DID WC CHANGE
003014  001401                          BEQ     .+4             ;NO
003016  104010                          HLT     !WC             ;WHY DID IT CHANGE?
```

```
1144                                    ;*************************************************************************
1145                                    ;TEST 3          SET AND CLEAR ALL REGISTERS
1146                                    ;*************************************************************************
1147  003020  104400                    TST3:   SCOPE
1148                                    ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
1149                                    ;BITS 7,6,5,4,3,2&1
1150
1151  003022  104414                            CLRDK                   ;CLEAR ALL RS REG
1152  003024  016767  176142  021462            MOV     TIMSV,TIMES     ;GET TIME
1153  003032  012777  003576  176040            MOV     #3576,@RSCS1    ;SET DISK FUNCTION BITS
1154  003040  022777  005776  176032            CMP     #5776,@RSCS1    ;ARE THESE BITS SET?
1155  003046  001401                            BEQ     .+4             ;NO
1156  003050  104001                            HLT     !CS1            ;SHOULD = 3776
1157  003052  012777  002524  176020            MOV     #2524,@RSCS1    ;SET THESE BITS
1158  003060  022777  004724  176012            CMP     #4724,@RSCS1    ;DID THEY SET
1159  003066  001401                            BEQ     .+4             ;YES
1160  003070  104001                            HLT     !CS1            ;SHOULD BE 2725
1161  003072  012777  001052  176000            MOV     #1052,@RSCS1    ;SET THESE BITS
1162  003100  022777  005252  175772            CMP     #5252,@RSCS1    ;ARE THEY =?
1163  003106  001401                            BEQ     .+4             ;YES
1164  003110  104001                            HLT     !CS1            ;SHOULD = 1252
1165  003112  104400                    TST4:   SCOPE
1166                                    ;CLEAR THE FUNCTION BITS
1167
1168  003114  012777  043576  175756            MOV     #43576,@RSCS1   ;SET DISK FUNCTION BITS
1169  003122  005077  175752                    CLR     @RSCS1
1170  003126  022777  004200  175744            CMP     #4200,@RSCS1    ;IS THE READY BIT SET
1171  003134  001401                            BEQ     .+4             ;YES
1172  003136  104001                            HLT     !CS1            ;RSCS1 SHOULD = 4200
1173
1174                                    ;*************************************************************************
1175                                    ;TEST 5          TEST RSCS2
1176                                    ;*************************************************************************
1177  003140  104400                    TST5:   SCOPE
1178
1179  003142  000005                            RESET                   ;CLEAR WORLD
1180  003144  022777  000100  175730            CMP     #100,@RSCS2     ;DID THEY CLEAR?
1181  003152  001401                            BEQ     .+4             ;YES
1182  003154  104200                            HLT     !CS2            ;NO
1183  003156  012777  021037  175716            MOV     #21037,@RSCS2   ;SET BITS 21017
1184  003164  022777  000137  175710            CMP     #137,@RSCS2     ;DID THESE BITS GET SET
1185  003172  001405                            BEQ     1$              ;YES
1186  003174  017700  175702                    MOV     @RSCS2,BAD
1187  003200  012701  000137                    MOV     #137,GOOD       ;WHAT CS2 SHOULD =
1188  003204  104000                            HLT                     ;CS2 = BAD GOOD = CORRECT ANS
```

```
1189  003206  012777  020025  175666  1$:     MOV     #20025,@RSCS2   ;SET THESE BITS
1190  003214  022777  000125  175660          CMP     #125,@RSCS2     ;DID THESE BITS GET SET
1191  003222  001401                          BEQ     .+4             ;YES
1192  003224  104200                          HLT     !CS2            ;NO,CS2 SHOULD = 20125
1193  003226  012777  000012  175646          MOV     #12,@RSCS2      ;LOAD THESE BITS
1194  003234  022777  000112  175640          CMP     #112,@RSCS2     ;DID THESE BITS GET SET IN CS2
1195  003242  001401                          BEQ     .+4             ;YES
1196  003244  104200                          HLT     !CS2            ;BAD = CS2 GOOD = CORRECT ANS
1197  003246  012777  177777  175626          MOV     #-1,@RSCS2      ;SET BITS
1198  003254  005077  175622                  CLR     @RSCS2          ;CLEAR THEM
1199  003260  022777  000100  175614          CMP     #100,@RSCS2     ;DID CLEAR WORK
1200  003266  001401                          BEQ     .+4             ;YES
1201  003270  104200                          HLT     !CS2            ;R/W BITS DID NOT CLEAR
1202  003272  016777  175664  175602          MOV     UNNUM,@RSCS2    ;GET UNIT #
1203  003300  104400                  TST6:   SCOPE
1204                                  ;CAN WE SET ALL THE RSBA BITS
1205
1206  003302  012777  177777  175576          MOV     #177777,@RSBA   ;SET THE BITS
1207  003310  022777  177776  175570          CMP     #177776,@RSBA   ;DID THEY SET
1208  003316  001401                          BEQ     .+4             ;YES
1209  003320  104020                          HLT     !BA             ;BITS 17776 SHOULD BE SET
1210  003322  012777  125252  175556          MOV     #125252,@RSBA   ;SET THESE BITS
1211  003330  022777  125252  175550          CMP     #125252,@RSBA   ;ARE THEY =
1212  003336  001401                          BEQ     .+4             ;YES
1213  003340  104020                          HLT     !BA             ;SHOULD BE 125252
1214  003342  012777  052524  175536          MOV     #52524,@RSBA    ;SET THESE BITS
1215  003350  022777  052524  175530          CMP     #52524,@RSBA    ;ARE THEY =
1216  003356  001401                          BEQ     .+4             ;YES
1217  003360  104020                          HLT     !BA             ;SHOULD BE 52524
1218
1219  003362  104400                  TST7:   SCOPE
1220                                  ;FLOAT A 1 THROUGH RSBA
1221
1222  003364  012701  000002          FLOTBA: MOV     #2,GOOD         ;GET A 2
1223  003370  000241                          CLC                     ;CLEAR CARRY
1224  003372  010177  175510          1$:     MOV     GOOD,@RSBA      ;FLOAT NUMBER
1225  003376  017700  175504                  MOV     @RSBA,BAD       ;GET BA
1226  003402  020100                          CMP     GOOD,BAD        ;COMPARE BA
1227  003404  001401                          BEQ     .+4             ;BA CORRECT
1228  003406  104000                          HLT                     ;BAD=BA GOOD=CORRECT ANS
1229  003410  006101                          ROL     GOOD            ;ROTATE NUMBER
1230  003412  103367                          BCC     1$              ;LOOP TILL DONE
```

```
1231  003414  104400                    TST10:  SCOPE
1232
1233                                    ;CLEAR THE RSBA REGISTER
1234
1235  003416  012777  177777  175462            MOV     #177777,@RSBA   ;SET RSBA EQUAL TO ALL ONES
1236  003424  005077  175456                    CLR     @RSBA
1237  003430  005777  175452                    TST     @RSBA           ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1238  003434  001401                            BEQ     .+4             ;YES
1239  003436  104020                            HLT     !BA             ;NO
1240  003440  104400                    TST11:  SCOPE
1241
1242                                    ;CAN WE SET ALL BITS IN RSWC REGISTER
1243
1244  003442  012777  177777  175434            MOV     #177777,@RSWC   ;SET WC BITS
1245  003450  022777  177777  175426            CMP     #177777,@RSWC   ;ARE ALL BITS SET
1246  003456  001401                            BEQ     .+4             ;YES
1247  003460  104010                            HLT     !WC             ;NO
1248  003462  012777  125252  175414            MOV     #125252,@RSWC   ;SET THESE BITS
1249  003470  022777  125252  175406            CMP     #125252,@RSWC   ;ARE THEY =
1250  003476  001401                            BEQ     .+4             ;YES
1251  003500  104010                            HLT     !WC             ;SHOULD BE 125252
1252  003502  012777  052525  175374            MOV     #52525,@RSWC    ;SET THESE BITS
1253  003510  022777  052525  175366            CMP     #52525,@RSWC    ;ARE THEY =
1254  003516  001401                            BEQ     .+4             ;YES
1255  003520  104010                            HLT     !WC             ;SHOULD BE 152525
1256  003522  104400                    TST12:  SCOPE
1257
1258                                    ;FLOAT A 1 THROUGH RSWC
1259
1260  003524  012701  000001            FLOTWC: MOV     #1,GOOD         ;GET A 1
1261  003530  000241                            CLC                     ;CLEAR CARRY
1262  003532  010177  175346            1$:     MOV     GOOD,@RSWC      ;FLOAT NUMBER
1263  003536  017700  175342                    MOV     @RSWC,BAD       ;GET WC
1264  003542  020100                            CMP     GOOD,BAD        ;COMPARE WC
1265  003544  001401                            BEQ     .+4             ;WC CORRECT
1266  003546  104000                            HLT                     ;BAD=WC GOOD=CORRECT ANS
1267  003550  006101                            ROL     GOOD            ;ROTATE NUMBER
1268  003552  103367                            BCC     1$              ;LOOP TILL DONE
```

```
1269                                    ;CLEAR THE WORD COUNT REGISTER
1270  003554  104400                    TST13:  SCOPE
1271
1272  003556  012777  177777  175320            MOV     #177777,@RSWC   ;SET RSWC REGISTER EQUAL TO ALL ONES
1273  003564  005077  175314                    CLR     @RSWC
1274  003570  005777  175310                    TST     @RSWC           ;DID ALL BITS GET CLEARED
1275  003574  001401                            BEQ     .+4             ;YES
1276  003576  104010                            HLT     !WC             ;NO
1277  003600  104400                    TST14:  SCOPE
1278
1279                                    ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
1280
1281  003602  012777  177777  175300            MOV     #177777,@RSDA   ;SET ALL BITS
1282  003610  022777  177777  175272            CMP     #177777,@RSDA   ;ARE THE BITS SET
1283  003616  001401                            BEQ     .+4             ;YES
1284  003620  104004                            HLT     !DA             ;NO
1285  003622  012777  125252  175260            MOV     #125252,@RSDA   ;SET THESE BITS
1286  003630  022777  125252  175252            CMP     #125252,@RSDA   ;ARE THEY =
1287  003636  001401                            BEQ     .+4             ;YES
1288  003640  104004                            HLT     !DA             ;SHOULD BE 125252
1289  003642  012777  052525  175240            MOV     #52525,@RSDA    ;SET THESE BITS
1290  003650  022777  052525  175232            CMP     #52525,@RSDA    ;ARE THEY =
1291  003656  001401                            BEQ     .+4             ;YES
1292  003660  104004                            HLT     !DA             ;SHOULD BE 52525
1293  003662  104400                    TST15:  SCOPE
1294
1295                                    ;FLOAT A 1 THROUGH RSDA
1296
1297  003664  012701  000001            FLOTDA: MOV     #1,GOOD         ;GET A 1
1298  003670  000241                            CLC                     ;CLEAR CARRY
1299  003672  010177  175212            1$:     MOV     GOOD,@RSDA      ;FLOAT NUMBER
1300  003676  017700  175206                    MOV     @RSDA,BAD       ;GET DA
1301  003702  020100                            CMP     GOOD,BAD        ;COMPARE DA
1302  003704  001401                            BEQ     .+4             ;DA CORRECT
1303  003706  104000                            HLT                     ;BAD=DA GOOD=CORRECT ANS
1304  003710  006101                            ROL     GOOD            ;ROTATE NUMBER
1305  003712  103367                            BCC     1$              ;LOOP TILL DONE
```

```
1306                                        ;CAN WE CLEAR THE RSDA REG.
1307  003714  104400                TST16:  SCOPE
1308
1309  003716  012777  177777  175164        MOV     #177777,@RSDA   ;SET RSDA TO ALL ONES
1310  003724  005077  175160                CLR     @RSDA
1311  003730  005777  175154                TST     @RSDA           ;TEST FOR ZERO RSDA
1312  003734  001401                        BEQ     .+4             ;YES
1313  003736  104004                        HLT     !DA             ;ANS SHOULD BE 0
1314  003740  104400                TST17:  SCOPE
1315
1316                                        ;SET AND CLEAR THE RSER REG.
1317
1318  003742  012777  177017  175144        MOV     #177017,@RSER   ;SET THESE BITS
1319  003750  022777  177017  175136        CMP     #177017,@RSER   ;DID THEY SET
1320  003756  001401                        BEQ     .+4             ;YES
1321  003760  104002                        HLT     !ER             ;RSER SHOULD = 157017
1322  003762  112777  000001  175124        MOVB    #1,@RSER        ;A MOVB INST
1323  003770  022777  000001  175116        CMP     #1,@RSER        ;SHOULD MODIFY COMPLETE WD
1324  003776  001401                        BEQ     .+4             ;OK
1325  004000  104002                        HLT     !ER
1326
1327  004002  104400                TST20:  SCOPE
1328
1329  004004  012777  052005  175102        MOV     #52005,@RSER    ;SET THESE BITS
1330  004012  022777  052005  175074        CMP     #52005,@RSER    ;DID THEY SET
1331  004020  001401                        BEQ     .+4             ;YES
1332  004022  104002                        HLT     !ER             ;ER SHOULD = 52005
1333  004024  104400                TST21:  SCOPE
1334
1335  004026  012777  125012  175060        MOV     #125012,@RSER   ;SET THESE BITS
1336  004034  022777  125012  175052        CMP     #125012,@RSER   ;DID THEY SET
1337  004042  001401                        BEQ     .+4             ;YES
1338  004044  104002                        HLT     !ER             ;ER SHOULD = 105012
```

```
1339  004046  104400                    TST22:  SCOPE
1340
1341  004050  012777  177017  175036            MOV     #177017,@RSER   ;SET THESE BITS
1342  004056  005077  175032                    CLR     @RSER           ;CLEAR THEM
1343  004062  005777  175026                    TST     @RSER           ;DID THEY CLEAR
1344  004066  001401                            BEQ     .+4             ;YES
1345  004070  104002                            HLT     !ER             ;SHOULD = 0
1346  004072  104400                    TST23:  SCOPE
1347
1348                                    ;SET AND CLEAR RSMR
1349
1350  004074  012777  000070  175022            MOV     #70,@RSMR       ;SET THESE BITS
1351  004102  017767  175016  175106            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
1352  004110  042767  177700  175100            BIC     #177700,WORK    ;CLEAR JUNK
1353  004116  022767  000070  175072            CMP     #70,WORK        ;DID THEY SET
1354  004124  001401                            BEQ     .+4             ;YES
1355  004126  104220                            HLT     !MR             ;SHOULD = 70
1356  004130  104400                    TST24:  SCOPE
1357
1358  004132  012777  000070  174764            MOV     #70,@RSMR       ;SET BITS
1359  004140  005077  174760                    CLR     @RSMR           ;CLEAR THEM
1360  004144  032777  000077  174752            BIT     #77,@RSMR       ;DID THEY CLEAR
1361  004152  001401                            BEQ     .+4             ;YES
1362  004154  104220                            HLT     !MR             ;BITS (77) SHOULD = 0
1363  004156  104400                    TST25:  SCOPE
1364
1365  004160  012777  000050  174736            MOV     #50,@RSMR       ;SET BITS
1366  004166  017767  174732  175022            MOV     @RSMR,WORK      ;PUT IN WORKABLE REG
1367  004174  042767  177700  175014            BIC     #177700,WORK    ;CLEAR JUNK
1368  004202  022767  000050  175006            CMP     #50,WORK        ;DID THESE BITS SET
1369  004210  001401                            BEQ     .+4             ;YES
1370  004212  104220                            HLT     !MR             ;BITS (50, SHOULD BE SET
1371  004214  104400                    TST26:  SCOPE
1372
1373  004216  012777  000020  174700            MOV     #20,@RSMR       ;SET BITS
1374  004224  017767  174674  174764            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
1375  004232  042767  177700  174756            BIC     #177700,WORK    ;CLEAR JUNK
1376  004240  022767  000020  174750            CMP     #20,WORK        ;DID THEY SET
1377  004246  001401                            BEQ     .+4             ;YES
1378  004250  104220                            HLT     !MR             ;MR SHOULD AT LEAST HAVE A (21)
```

```
                                ;*****************************************************************************
                                ;TEST 27                    TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
                                ;*****************************************************************************
004252  104400            TST27:  SCOPE

004254  104414            BITST:  CLRDK                           ;CLEAR ALL RS REG
004256  012777  003566  174614    MOV     #3566,@RSCS1            ;LOAD CS1
004264  112777  000005  174642    MOVB    #5,@RSCS1B              ;LOAD BIT
004272  022777  004766  174600    CMP     #4766,@RSCS1            ;DID IT LOAD?
004300  001401                    BEQ     .+4                     ;YES
004302  104001                    HLT     !CS1
004304  112777  000032  174566    MOVB    #32,@RSCS1
004312  022777  004632  174560    CMP     #4632,@RSCS1
004320  001401                    BEQ     .+4
004322  104001                    HLT     !CS1                    ;CS1 SHOULD = 6632

004324  104400            TST30:  SCOPE

004326  016777  174630  174546  BITCS2: MOV     UNNUM,@RSCS2            ;LOAD UNIT NUMBER
004334  052777  177400  174540          BIS     #177400,@RSCS2          ;LOAD ALL BITS
004342  105077  174570                  CLRB    @RSCS2B                 ;CLR UPPER BYTE
004346  016701  174610                  MOV     UNNUM,GOOD              ;GET UNIT NO.
004352  052701  000100                  BIS     #100,GOOD               ;SET OR BIT
004356  017700  174520                  MOV     @RSCS2,BAD              ;GET CS2
004362  020001                          CMP     BAD,GOOD                ;IS CS2 CORRECT?
004364  001401                          BEQ     .+4                     ;YES
004366  104000                          HLT                             ;LOAD BYTE DID NOT WORK

004370  104400            TST31:  SCOPE

004372  012777  025252  174504  BITWC:  MOV     #25252,@RSWC            ;LOAD WC
004400  112777  000377  174532          MOVB    #377,@RSWCB             ;LOAD BIT
004406  022777  177652  174470          CMP     #177652,@RSWC           ;DID IT LOAD?
004414  001401                          BEQ     .+4                     ;YES
004416  104010                          HLT     !WC                     ;NO WC SHOULD =177652
004420  112777  000123  174456          MOVB    #123,@RSWC
004426  022777  177523  174450          CMP     #177523,@RSWC
004434  001401                          BEQ     .+4
004436  104010                          HLT     !WC                     ;WC SHOULD = 177523

004440  104400            TST32:  SCOPE

004442  012777  025252  174436  BITBA:  MOV     #25252,@RSBA            ;LOAB DA
004450  112777  000377  174464          MOVB    #377,@RSBAB             ;LOAD BIT
004456  022777  177652  174422          CMP     #177652,@RSBA           ;DID IT LOAD?
004464  001401                          BEQ     .+4                     ;YES
004466  104020                          HLT     !BA                     ;DA SHOULD =177652
004470  112777  000125  174410          MOVB    #125,@RSBA
004476  022777  177524  174402          CMP     #177524,@RSBA
004504  001401                          BEQ     .+4
004506  104020                          HLT     !BA                     ;BA SHOULD = 177525
004510  104414                          CLRDK                           ;CLEAR ALL RS REG
```

```
                                ;*****************************************************************************
                                ;TEST 33                    LOAD RSDB WITH ALL ONES AND ALL ZEROS
                                ;*****************************************************************************
004512  104400                  TST33:  SCOPE

004514  104414                  ZERONE: CLRDK                           ;CLEAR ALL RS REG
004516  005077  174400                  CLR     @RSDB                   ;LOAD DB WITH ALL 0
004522  012777  177777  174372        MOV     #177777,@RSDB           ;LOAD DB WITH ALL ONES
004530  012767  002000  174460        MOV     #2000,WORK              ;TIME OUT ROUTINE
004536  012701  000300                MOV     #300,GOOD               ;GET CORRECT FOR CS2
004542  056701  174414                BIS     UNNUM,GOOD
004546  017700  174330          2$:   MOV     @RSCS2,BAD              ;GET CS2
004552  020100                      CMP     GOOD,BAD                ;IS IT CORRECT?
004554  001404                      BEQ     3$                      ;YES
004556  005367  174434              DEC     WORK                    ;TO WAIT FOR OR
004562  001371                      BNE     2$                      ;TO SET
004564  104200                      HLT     !CS2                    ;OR SHOULD BE SET
004566  005001                  3$: CLR     GOOD
004570  017700  174326              MOV     @RSDB,BAD               ;LOAD BAD WITH DB
004574  020100                      CMP     GOOD,BAD                ;IS BAD CORRECT
004576  001401                      BEQ     .+4                     ;YES
004600  104000                      HLT                             ;COULD NOT FLOAT 0 THROUGH DB
004602  012701  177777              MOV     #-1,GOOD                ;LOAD GOOD WITH ANS
004606  017700  174310              MOV     @RSDB,BAD               ;GET DATA FROM DB
004612  020100                      CMP     GOOD,BAD                ;IS DB CORRECT
004614  001401                      BEQ     .+4                     ;YES
004616  104000                      HLT                             ;BAD SHOULD = 1777777
                                ;TEST INTERRUPT IN THE RH11
                                ;BY MOVING 300 INTO RHCS1
                                ;*****************************************************************************
                                ;TEST 34                    TEST INTERRUPT IN RH11
                                ;*****************************************************************************
004620  104400                  TST34:  SCOPE
004622  104414                  INT:    CLRDK                           ;CLEAR ALL ERRORS
004624  012777  004676  174276        MOV     #PGTRAP,@RSVEC          ;SET UP VECTOR
004632  012777  000340  174272        MOV     #340,@RSVCPS            ;SET TRAP PS
004640  012737  000200  177776        MOV     #200,@#PS               ;SET PS AT PRIORITY 4
004646  012777  000300  174224        MOV     #300,@RSCS1             ;THIS SHOULD CAUSE A TRAP
004654  012767  000500  174334        MOV     #500,WORK               ;SETUP LOOP
004662  005367  174330          1$:   DEC     WORK                    ;DEC LOOP SHOULD
004666  001375                      BNE     1$                      ;INTERRUPT BEFORE LOOP IS DONE
004670  104001                      HLT     !CS1                    ;SHOULD NEVER GET HERE
004672  000167  000014              JMP     INTDON                  ;GET OUT
004676  022626                  PGTRAP: CMP     (6)+,(6)+               ;TRAP OK
004700  022777  004200  174172        CMP     #4200,@RSCS1            ;DID IE CLEAR?
004706  001401                      BEQ     .+4                     ;YES
004710  104001                      HLT     !CS1                    ;IE SHOULD BE CLEARED
004712                          INTDON:
```

```
                                ;*******************************************************************************
                                ;TEST 35                    MAINTENANCE TIMING TEST
                                ;*******************************************************************************
004712  104400                  TST35:  SCOPE

                                ;MODULE TESTED G092
                                ;THE FOLLOWING TEST ON THE RS03 DISK IS A SINGLE-STEPPED
                                ;MAINTENANCE MODE TEST ON THE RS03 TIMING LOGIC.  THE ACTUAL
                                ;DISK SURFACE IS SUBSTITUTED BY THE MAINTENANCE RESISTER--I.E.
                                ;THE PROGRAM WILL SUPPLY ALL "DISK CLOCK" PULSES TO DRIVE THE
                                ;TIMING LOGIC.  WE ARE TESTING THE ENTIRE TIMING TRACK LOGIC, INCLUDING INDEX,
                                ;PULSE FUNCTION, RESYNC AREA, SECTOR COUNTERS, ETC.

                                ;PUT DRIVE IN MAINTENANCE MODE
004714  104414                  MRTIME: CLRDK                   ;CLEAR DRIVE REGISTERS
004716  052767  001040  174244          BIS     #1040,ONCEE     ;SET CLK CNT
004724  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
004726  104420                          MRCK                    ;CHECK MAINTENANCE REG FOR
004730  022701                          22701                   ;22701
004732  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
004734  104430                          MRIND                   ;SEND MAINT INDEX PULSE

004736  104420                          MRCK                    ;CHECK MAINT REG TO
004740  022701                          22701                   ;EQUAL 22701
004742  104000                          HLT                     ;MR=BAD GOOD=CORRECTIONS
                                                                ;COULD NOT INITIALIZE MR REG
                                ;INDEX PULSE SHOULD CLEAR LOOK-AHEAD REG

004744  005777  174150                  TST     @RSLA           ;IS RSLA CLEARED
004750  001401                          BEQ     .+4             ;YES
004752  104224                          HLT     !MR!LA          ;RSLA SHOULD BE CLEARED
                                                                ;WITH THE INDEX PULSE

                                ;PERFORM MAINTENANCE CLOCK OPERATION 512 TIMES TO
                                ;PROVIDE CLOCK TO STEP TIMING THRU RESYNC PERIOD
                                ;IF SECTOR PULSE IS ASSERTED DURING THIS LOOP
                                ;CHECK SECTOR BOUNDARY COUNTER AND E12

004754  012767  001000  174222          MOV     #512.,REPT
004762  104446  MRTIM1: MCLK1                                   ;CLOCK MAINT REG WITH AN 11
004764  104420                          MRCK                    ;CHECK MR REG TO
004766  032711                          32711                   ;EQUAL 32711
004770  104000                          HLT                     ;MR = BAD GOOD = CORRECT ANS
004772  104450                          MCLK0                   ;CLOCK MR WITH A 1
004774  104420                          MRCK                    ;CHECK MR TO
004776  022701                          22701                   ;EQUAL 22701
005000  104000                          HLT                     ;BAD=MR REG GOOD=CORRECTIONS
005002  005367  174176                  DEC     REPT            ;IS THE LOOP DONE YET?
005006  001365                          BNE     MRTIM1          ;NO-LOOP
```

```
1529                                    ;AFTER ONE MORE CLOCK, SECTOR PULSE SHOULD BE ASSERTED
1530                                    ;IF NOT, CHECK SECTOR BOUNDARY COUNTER, SECTOR BOUNDARY FF (E21) AND E12
1531
1532  005010  104446                            MCLK1                   ;CLOCK MAINT REG WITH AN 11
1533  005012  104420                            MRCK                    ;CHECK MR REG TO
1534  005014  032311                            32311                   ;EQUAL 32311
1535  005016  104000                            HLT                     ;MR=BAD GOOD=CORRECTIONS
1536  005020  104450                            MCLKO                   ;CLOCK MR WITH A 1
1537  005022  104420                            MRCK                    ;CHECK MAINT REG
1538  005024  022301                            22301                   ;TO EQUAL 22301
1539  005026  104000                            HLT                     ;MR=BAD GOOD-CORRECT ANS
1540  005030  005777  174064                   TST     @RSLA           ;DOES LOOK AHEAD REG=0
1541  005034  001401                            BEQ     MRT2            ;YES-CONT
1542  005036  104224                            HLT     !MR!LA          ;LOOK AHEAD REG SHOULD=0
1543                                    ;PERFORM MAINTENANCE CLOCK OPERATION 40 TIMES TO PROVIDE
1544                                    ;CLOCK PULSES TO STEP THRU 1ST SECTOR PRE-AMBLE AREA
1545
1546  005040  005002                    MRT2:   CLR     R2              ;CLEAR R2 FOR SECTOR COMPARE WITH LA REG
1547  005042  012767  000050  174134            MOV     #40.,REPT       ;40 CLOCKS TO STEP THRU PRE-AMBLE
1548  005050  104446                    MRT2A:  MCLK1                   ;CLOCK MR WITH AN 11
1549  005052  104420                            MRCK                    ;CHECK MAINT REG
1550  005054  033711                            33711                   ;EQUAL 33711
1551  005056  104000                            HLT                     ;MR = BAD GOOD = CORRECT ANS
1552  005060  104450                            MCLKO                   ;CLOCK MR REG WITH A 1
1553  005062  104420                            MRCK                    ;CHECK MR REG
1554  005064  023701                            23701                   ;TO EQUAL 23701
1555  005066  104000                            HLT                     ;MR = BAD GOOD = CORRECTANS
1556  005070  005367  174110                   DEC REPT                ;REPEAT
1557  005074  001365                            BNE     MRT2A           ;LOOP 40 TIMES
1558
1559                                    ;SUPPLY CLOCKS TO STEP THROUGH THE DATA AREA IN THE SECTOR
1560  005076  012767  002200  174100            MOV     #18.*64.,REPT   ;18 CLOCKS PER DATA WORD
1561  005104  104446                    MRT2B:  MCLK1                   ;CLOCK MR WITH AN 11
1562  005106  104420                            MRCK                    ;CHECK MAINT REG
1563  005110  033711                            33711                   ;TO EQUAL 33711
1564  005112  104000                            HLT                     ;MR = BAD GOOD = CORRECT ANS
1565  005114  104450                            MCLKO                   ;CLOCK MR REG WITH A 1
1566  005116  104420                            MRCK                    ;CHECK MR REG
1567  005120  023701                            23701                   ;TO EQUAL 23701
1568  005122  104000                            HLT                     ;MR=BAD GOOD=CORRECTANS
1569  005124  005367  174054                   DEC     REPT            ;REPEAT
1570  005130  001365                            BNE     MRT2B           ;LOOP
```

```
                                        ;SUPPLY ENOUGH MAINT CLOCKS TO STEP THROUGH THE CRC AREA
                                        ;AND THE DEAD BAND ON THE SECTOR
005132  012767  000214  174044          MOV     #140.,REPT      ;AMOUNT OF CLOCKS TO END OF SECTOR
005140  104446                  MRT2C:  MCLK1                   ;CLOCK MR WITH AN 11
005142  104420                  MRCK                            ;CHECK MAINT REG
005144  033711                  33711                           ;TO EQUAL 33711
005146  104000                  HLT                             ;MR = BAD GOOD = CORRECT ANS
005150  104450                  MCLK0                           ;CLOCK MR REG WITH A 1
005152  104420                  MRCK                            ;CHECK MAINT REG
005154  023701                  23701                           ;TO EQUAL 23701
005156  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
005160  005367  174020          DEC     REPT                    ;REPEAT
005164  001365                  BNE     MRT2C                   ;LOOP
005166  104446                  MCLK1                           ;CLOCK MR REG WITH 11
005170  104420                  MRCK                            ;CHECK MR REG
005172  033711                  33711                           ;TO EQUAL 33711
005174  104000                  HLT                             ;MR = BAD GOOD = CORRECT ANS
                                        ;ONE MORE CLOCK SHOULD CAUSE SECTOR PULSE
                                        ;IF NOT, CHECK E16-6

005176  104450                  MCLK0                           ;CLOCK MR WITH A 1
005200  104420                  MRCK                            ;MAINT REG SHOULD
005202  023701                  23701                           ;EQUAL 23701
005204  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
005206  104446                  MCLK1                           ;CLOCK MR WITH AN 11
005210  104420                  MRCK                            ;CHECK MAINT REG
005212  032311                  32311                           ;SHOULD EQUAL 32311
005214  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS

                                        ;LOOK-AHEAD REGISTER SHOULD NOW POINT TO SECTOR 1 (OR 4000 IF INTERLEAVED)

005216  022777  000000  173702          CMP     #0,@RSDT        ;INTERLEAVED?
005224  001403                  BEQ     3$                      ;NO
005226  062702  004000                  ADD     #4000,R2        ;YES
005232  000402                  BR      2$                      ;CONT
005234  062702  000100          3$:     ADD     #100,R2         ;INCREMENT SECTOR COMPARE
005240  020277  173654          2$:     CMP     R2,@RSLA        ;LA REG SHOULD=100
005244  001401                  BEQ     1$                      ;LA IS CORRECT
005246  104224                  HLT     !MR!LA                  ;LA SHOULD=100
```

```
                                        ;REPEAT NEXT STEPS 62 TIMES.  LOOK-AHEAD REGISTER SHOULD INCREMENT
                                        ;TO SHOW NEXT SECTOR, CHECKS FOR ALL SECTORS. IF DRIVE IS NOT
                                        ;INTERLEAVED, LA = 200,300, ETC. IF DRIVE IS INTERLEAVED,
                                        ;LA = 100, 4100, 200, 4200 ETC. SEE SERVICE MANUAL FOR DETAILS.

005250  012767  000076  173730  1$:     MOV     #62.,REPT1
005256  012767  002465  173720  MRT3:   MOV     #1333.,REPT
005264  104452                  3$:     MCLKB                   ;CLOCK MR WITH A 1 AND A 11
005266  005367  173712                  DEC     REPT            ;STEP THROUGH
005272  001374                          BNE     3$              ;SECTOR
005274  104450                          MCLKO                   ;CLOCK MR WITH A 1
005276  104420                          MRCK                    ;MAINT REG
005300  022701                          22701                   ;SHOULD EQUAL 22701
005302  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
005304  104446                          MCLK1                   ;1 MORE CLK ASSERTS SECTOR PULSE
005306  104420                          MRCK                    ;MAINT REG SHOULD
005310  032311                          32311                   ;EQUAL 32311
005312  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
005314  022777  000000  173604          CMP     #0,@RSDT        ;DRIVE INTERLEAVED?
005322  001420                          BEQ     6$              ;YES
005324  032767  001000  173636          BIT     #BIT9,ONCEE     ;DO I SET 4000
005332  001406                          BEQ     4$              ;OR CLEAR IT IN RSLA
005334  042767  001000  173626          BIC     #BIT9,ONCEE
005342  162702  004000                  SUB     #4000,R2
005346  000406                          BR      6$
005350  052767  001000  173612  4$:     BIS     #BIT9,ONCEE
005356  062702  004000                  ADD     #4000,R2
005362  000402                          BR      5$
005364  062702  000100          6$:     ADD     #100,R2         ;INCREMENT SECTOR COMPARE
005370  017700  173524          5$:     MOV     @RSLA,BAD       ;LA REG SHOULD HAVE INCREMENTED TO NEXT SECTOR
005374  010201                          MOV     R2,GOOD         ;GET CORRECT ANS FOR RSLA
005376  020100                          CMP     GOOD,BAD        ;COMPARE FOR CORRECT ANS
005400  001401                          BEQ     1$              ;RSLA IS GOOD
005402  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS

005404  005367  173576          1$:     DEC     REPT1           ;REPEAT 62
005410  001322                          BNE     MRT3            ;TIMES
005412  012767  002465  173564          MOV     #1333.,REPT     ;COUNT FOR LAST SECTOR
005420  104452                  2$:     MCLKB                   ;CLOCK
005422  005367  173556                  DEC     REPT            ;THRU
005426  001374                          BNE     2$              ;LAST SECTOR
005430  017700  173464                  MOV     @RSLA,BAD       ;GET CONTENTS OF RSLA
005434  012701  007777                  MOV     #7777,GOOD      ;GET CORRECT ANS
005440  020100                          CMP     GOOD,BAD        ;DOES RSLA EQUAL 7777
005442  001401                          BEQ     .+4             ;YES
005444  104000                          HLT                     ;BAD=RSLA GOOD=CORRECT ANS
```

```
                                ;*****************************************************************************
                                ;TEST 36                    SECTOR FRACTION TEST
                                ;*****************************************************************************
005446  104400                  TST36:  SCOPE

                                ;MODULE TESTED G092
                                ;CLOCK THROUGH AN ENTIRE TRACK IN MAINT MODE WHILE
                                ;CHECKING FOR THE PROPER OPERATION OF THE SECTOR FRACTION IN THE LOOK-AHEAD REG.
                                ;WHEN THE LAST WORD IS BEING TRANSFERRED, SECTOR AND FRACTION
                                ;IS EQUAL TO 7777 TO INDICATE LAST WORD ON THIS TRACK --
                                ;HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER WILL
                                ;CLEAR THE FRACTION BITS IF ANOTHER WORD IS CLOCKED. RSLA
                                ;SHOULD INDICATE 7700 ON ANOTHER MAINTENANCE CLOCK.

005450  104414                  MRT4:   CLRDK                   ;CLEAR DRIVE REGISTERS
005452  052767  000040  173510          BIS     #40,ONCEE       ;SET FLAG BITS
005460  042767  003000  173502          BIC     #3000,ONCEE
005466  005067  173504                  CLR     MCCNT           ;CLEAR MAINT CLOCK COUNTER
005472  005002                          CLR     R2              ;CLEAR R2 FOR SECTOR COUNTER
005474  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
005476  104420                          MRCK                    ;CHECK MAINTENANCE REG FOR
005500  022701                          22701                   ;22701
005502  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
005504  104430                          MRIND                   ;ISSUE A MAINT INDEX PULSE
                                                                ;TO CLEAR THE DRIVE
005506  104420                          MRCK                    ;CHECK MAINT REG
005510  022701                          22701                   ;TO EQUAL 22701
005512  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;ISSUE 512 MAINT CLOCKS TO STEP THROUGH THE RESYNC AREA

005514  012767  001000  173462          MOV     #512.,REPT      ;COUNT TO STEP THRU RESYNC AREA
005522  104446          MRT4A:  MCLK1                   ;CLOCK THROUGH RESYNC
005524  104420                  MRCK                    ;CHECK MAINT REG
005526  032711                  32711                   ;TO EQUAL 32711
005530  104000                  HLT                     ;MR = BAD GOOD = CORRECT ANS
005532  104450                  MCLK0                   ;CLOCK MR REG
005534  104420                  MRCK                    ;CHECK MR REG
005536  022701                  22701                   ;TO EQUAL 22701
005540  104000                  HLT                     ;BAD=MR GOOD=CORRECT ANS
005542  022777  000000  173350          CMP     #0,@RSLA        ;LOOK AHEAD REG
005550  001401                  BEQ     .+4             ;EQUAL 0
005552  104204                  HLT     !LA
005554  005367  173424                  DEC     REPT            ;LOOP THROUGH
005560  001360                  BNE     MRT4A           ;RESYNC AREA

                                ;ONE MORE PULSE SHOULD CAUSE THE FIRST SECTOR PULSE

005562  104446                  MCLK1                   ;CLOCK MR WITH AN 11
005564  104420                  MRCK                    ;CHECK MAINT REG FOR SECTOR PULSE
005566  032311                  32311                   ;MR SHOULD=32311
005570  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                MRT4B:  MCLKD                   ;CLOCK MR REG WITH A 1
                                        MRCK                    ;CHECK MAINT REG
                                        22301                   ;TO EQUAL 22301
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;SECTOR FRACTION BITS IN LOOK-AHEAD REGISTER SHOULD BE CLEARED (EQUAL TO 00)

                                        MOV     @RSLA,BAD       ;GET RSLA
                                        MOV     R2,GOOD         ;GET CORRECT ANS
                                        CMP     GOOD,BAD        ;IS THE RSLA REG CORRECT
                                        BEQ     1$              ;YES
                                        HLT                     ;RSLA=BAD GOOD=CORRECTANS

                                ;STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA
                                ;AREA WHILE CHECKING THE SECTOR FRACTION

                        1$:     MOV     #82.,REPT       173360  ;FOR FIRST FRACTION CHANGE
                                MRCLK                           ;CLOCK MR REG WITH AN 11 AND A 1
                        MRT4C:  MOV     @RSLA,BAD       173266  ;GET RSLA
                                MOV     R2,GOOD                 ;GET CORRECT ANS
                                CMP     BAD,GOOD                ;IS RSLA CORRECT
                                BEQ     1$                      ;YES
                                HLT                             ;BAD=RSLA GOOD=CORRECT ANS
                        1$:     DEC     REPT            173336  ;LOOP ON
                                BNE     MRT4C                   ;PREAMBLE AREA

                                ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE

                                MRCLK                           ;CLOCK MR WITH AN 11 AND A 1
                                INC     R2                      ;COUNT THE FRACTION
                                MOV     @RSLA,BAD       173240  ;GET RSLA
                                MOV     R2,GOOD                 ;GET CORRECT ANS
                                CMP     BAD,GOOD                ;IS RSLA CORRECT?
                                BEQ     2$                      ;YES
                                HLT                             ;RSLA=BAD GOOD=CORRECT ANS

                                ;FIRST FRACTION CHANGES AFTER 82 MAINT CLKS, THE REST
                                ;CHANGE AFTER 20 MAINTENANCE CLOCKS

                        2$:     MOV     #62.,REPT       173306  ;COUNT FOR WORDS IN A SECTOR
                        MRT4D:  MOV     #19.,REPT1      173302  ;COUNT FOR SECT FRACT TO CHANGE
                        MRT4E:  MRCLK                           ;CLOCK MR WITH AN 11 AND A 1
                                MOV     @RSLA,BAD       173206  ;GET RSLA
                                MOV     R2,GOOD                 ;GET CORRECT ANS
                                CMP     GOOD,BAD                ;IS RSLA CORRECT?
                                BEQ     1$                      ;YES
                                HLT                             ;RSLA=BAD GOOD=CORRECT ANS
                        1$:     DEC     REPT1           173260  ;LOOP
                                BNE     MRT4E
```

```
                                        ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE

005730  104422                          MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
005732  022702  007777                  CMP     #7777,R2        ;AT THE LAST SECTOR-LAST FRACTION?
005736  001472                          BEQ     MRT4F           ;YES, FINISH THE SECTOR
005740  005202                          INC     R2              ;NO, ADD 1 TO FRACTION
005742  017700  173152          4$:     MOV     @RSLA,BAD       ;GET RSLA
005746  022777  000000  173152          CMP     #0,@RSDT        ;IS THIS DRIVE INTERLEAVED?
005754  001431                          BEQ     12$             ;NO
005756  032767  002000  173204          BIT     #BIT10,ONCEE    ;HAS REPT GONE TO ZERO YET FOR THIS SECTOR?
005764  001425                          BEQ     12$             ;NO
                                                                ;RSLA NOW POINTS TO NEXT INTERLEAVED
                                                                ;SECTOR; BIT 9 IN ONCEE INDICATES
                                                                ;WHETHER RSLA SHOULD NOW
                                                                ;BE BETWEEN 0000-3700(1)
                                                                ;OR 4000-7700(0).
005766  032767  001000  173174          BIT     #BIT9,ONCEE     ;SHOULD RSLA BE BETWEEN 0000-3700?
005774  001004                          BNE     9$              ;YES
005776  052767  001000  173164          BIS     #BIT9,ONCEE     ;SET FOR NEXT PASS
006004  000406                          BR      10$             ;
006006  042767  001000  173154  9$:     BIC     #BIT9,ONCEE     ;CLEAR FOR NEXT PASS
006014  042702  004000                  BIC     #4000,R2        ;MAKE RSLA LESS THAN 4000
006020  000404                          BR      5$
006022  062702  004000          10$:    ADD     #4000,R2        ;COMPENSATE FOR
006026  162702  000100                  SUB     #100,R2         ;INTERLEAVING
006032  042767  002000  173130  5$:     BIC     #BIT10,ONCEE    ;CLEAR FLAG FOR NEXT SECTOR
006040  010201                  12$:    MOV     R2,GOOD         ;GET CORRECT ANSWER FOR RSLA
006042  020100                          CMP     GOOD,BAD        ;IS RSLA CORRECT
006044  001401                          BEQ     2$              ;YES
006046  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS
006050  005367  173130          2$:     DEC     REPT            ;HAS SECTOR FRACTION REACHED 77?
006054  001310                          BNE     MRT4D           ;NO

                                        ;CHECK FOR END OF ONE SECTOR OR BEGINNING OF NEXT

006056  010203                  11$:    MOV     R2,R3
006060  042703  177700                  BIC     #177700,R3      ;CHECK SECTOR FRACTION
006064  022703  000077                  CMP     #77,R3          ;END OF SECTOR?
006070  001402                          BEQ     3$              ;YES
006072  000167  177474                  JMP     MRT4B           ;NO, BEGINNING OF NEXT
006076  012767  000012  173102  3$:     MOV     #10.,REPT1      ;SETUP LOOP TO FINISH
006104  012767  000001  173072          MOV     #1,REPT         ;THIS SECTOR
006112  052767  002000  173050          BIS     #BIT10,ONCEE    ;REPT HAS GONE TO ZERO FOR THIS SECTOR
006120  000167  177560                  JMP     MRT4E           ;LOOP

006124  012767  000010  173052  MRT4F:  MOV     #8.,REPT
006132  104422          1$:     MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
006134  017700  172760                  MOV     @RSLA,BAD       ;GET RSLA
006140  010201                          MOV     R2,GOOD         ;R2 SHOULD=7777
006142  020100                          CMP     GOOD,BAD        ;IS RSLA CORRECT-END OF DISK?
006144  001401                          BEQ     2$              ;YES
006146  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS (7777)
006150  005367  173030          2$:     DEC     REPT            ;FINISH
006154  001366                          BNE     1$              ;LOOP
```

```
1813                                    ;SECTOR AND FRACTION IS = TO 7777 TO INDICATE LAST WORD ON THIS TRACK
1814                                    ;RSLA SHOULD EQUAL 7700 ON ANOTHER MAINT CLOCK.
1815
1816  006156  104422            MRT4G:  MRCLK                   ;CLOCK MR WITH AN 11 AND A 1
1817  006160  017700  172734            MOV     @RSLA,BAD       ;GET RSLA
1818  006164  012701  007700            MOV     #7700,GOOD      ;GET CORRECT ANS
1819  006170  020100                    CMP     GOOD,BAD        ;IS RSLA CORRECT?
1820  006172  001401                    BEQ     1$              ;YES
1821  006174  104000                    HLT                     ;RSLA=BAD GOOD=CORRECT ANS
1822  006176  104430            1$:     MRIND                   ;ISSUE AN INDEX PULSE TO
1823                                                            ;CLEAR THE DRIVE
1824  006200  017700  172714            MOV     @RSLA,BAD       ;GET RSLA
1825  006204  005001                    CLR     GOOD            ;GET CORRECT ANS
1826  006206  020100                    CMP     GOOD,BAD        ;IS RSLA CORRECT?
1827  006210  001401                    BEQ     2$              ;YES
1828  006212  104000                    HLT                     ;RSLA=BAD GOOD=CORRECT ANS
1829  006214  104420            2$:     MRCK                    ;CHECK MR REG
1830  006216  022701                    22701                   ;TO EQUAL 22701
1831  006220  104000                    HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                ;*****************************************************************************
                                ;TEST 37                    ILLEGAL FUNCTION TEST
                                ;*****************************************************************************
006222  104400                  TST37:  SCOPE

                                ;MODULE TESTED M7759, M7770
                                ;TEST ILLEGAL FUNCTION (ILF) IN RSER. SEND AN ILLEGAL FUNCTION
                                ;CODE TO THE DRIVE CONTROL REGISTER WITHOUT SETTING THE GO BIT.
                                ;THE "ILF" BIT SHOULD NOT BE SET. THE "GO" BIT IS THEN SET.  A
                                ;CHECK IS THEN  MADE FOR "ATA" AND "ERR" TO BE SET
                                ;IN THE DRIVE STATUS REGISTER (RSDS) AND "ILF" IN THE DRIVE ERROR
                                ;REGISTER (RSER).  ALL ILLEGAL FUNCTION CODES ARE CHECKED.
                                ;ILLEGAL FUNCTIONS ARE DETECTED ON M7759 BY E20-8

006224  104414                  MRILF:  CLRDK                   ;CLEAR ALL THE DRIVE REGISTERS
006226  042767  000040  172734          BIC     #BIT5,ONCEE     ;CLEAR CLOCK CNT FLAG
006234  032767  000002  172726          BIT     #BIT1,ONCEE     ;WAS THERE AN ERROR
006242  001002                          BNE     MRLF1           ;YES DO NOT CHANGE "ILF" CODE
006244  012702  000003                  MOV     #3,R2           ;SETUP FIRST "ILF" CODE
                                ;PUT DRIVE IN MAINTENANCE MODE

006250  104416                  MRLF1:  MRDMD                   ;PUT DRIVE INTO MAINT MODE
006252  104420                          MRCK                    ;CHECK MR REG TO
006254  022701                          22701                   ;EQUAL 22701
006256  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)

                                ;ASSERT A MAINTENANCE MODE DISK "INDEX" PULSE

                                        MRIND
006260  104430                  MRLF2:  MOV     R2,@RSCS1       ;SEND "ILF" WITH THE "GO" BIT
006262  010277  172612                  MOV     @RSDS,BAD       ;GET DRIVE STATUS REG
006266  017700  172620                  MOV     #150600,GOOD    ;GET CORRECT ANS
006272  012701  150600                  CMP     GOOD,BAD        ;IS RSDS CORRECT?
006276  020100                          BEQ     1$              ;YES
006300  001440                          TYPE    .+2             ;.ASCIZ <15><12>"ILLEGAL FUNCTION CODE SENT TO DRIVE= "
006302  104402  006306                  MOV     R2,WORK         ;GET FUNCTION CODE
006306  010267  172634                  MOV     WORK,-(6)       ;PUT WORK ON STACK
006312  016746  172630                  TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
006316  104406                          BIS     #BIT1,ONCEE     ;SET ERROR BIT SO ILLEGAL FUN DOESN'T CHANGE
006320  052767  000002  172572          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
006326  104000                          HLT     !DS
006330  104040

006332  042767  000002  172560  1$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
006340  017700  172500                  MOV     @RSER,BAD       ;GET RSER
006344  012701  000001                  MOV     #1,GOOD         ;GET CORRECT ANS
006350  020100                          CMP     GOOD,BAD        ;DID "ILF" SET IN RSER
006352  001404                          BEQ     2$              ;YES
006354  052767  000002  172536          BIS     #BIT1,ONCEE     ;SET ERROR BIT
006362  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
006364  042767  000002  172526  2$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
```

```
                                        ;CLEAR THE DRIVE FOR THE NEXT "ILF" CODE PASS
006442  104414                  MRCILF: CLRDK                   ;CLEAR ERRORS
006444  017700  172442                  MOV     @RSDS,BAD       ;GET RSDS REG
006450  012701  010600                  MOV     #10600,GOOD     ;GET CORRECT ANS
006454  020100                          CMP     GOOD,BAD        ;DID "ATA" AND "ERR" CLEAR IN RSDS?
006456  001435                          BEQ     1$              ;YES
006460  104402  006464                  TYPE    ,.+2            ;.ASCIZ <15><12>"ATA AND ERR IN RSDS SHOULD CLEAR WITH I
006542  052767  000002  172420          BIS     #BIT1,ONCEE
006550  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
006552  042767  000002  172410  1$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
006560  017700  172330                  MOV     @RSER,BAD       ;GET RSER
006564  005001                          CLR     GOOD            ;GET CORRECT ANS
006566  020100                          CMP     GOOD,BAD        ;DID ILF CLEAR IN RSER
006570  001431                          BEQ     2$              ;YES
006572  052767  000002  172370          BIS     #BIT1,ONCEE     ;SET ERROR BIT
006600  104402  006604                  TYPE    ,.+2            ;.ASCIZ <15><12>"ILF IN RSER SHOULD CLEAR WITH INIT"
006652  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
006654  042767  000002  172306  2$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR BIT
                                        ;GET NEXT ILLEGAL FUNCTION COE

006662  062702  000002          MRLF3:  ADD     #2,R2           ;UPDATE ILF
006666  022702  000011                  CMP     #11,R2          ;IS THIS A ILF CODE
006672  001773                          BEQ     MRLF3           ;NO-UPDATE IT
006674  022702  000021                  CMP     #21,R2
006700  001770                          BEQ     MRLF3
006702  022702  000031                  CMP     #31,R2
006706  001765                          BEQ     MRLF3
006710  022702  000051                  CMP     #51,R2
006714  001762                          BEQ     MRLF3
006716  022702  000061                  CMP     #61,R2
006722  001757                          BEQ     MRLF3
006724  022702  000071                  CMP     #71,R2
006730  001754                          BEQ     MRLF3
006732  022702  000101                  CMP     #101,R2
006736  001402                          BEQ     ILFDON          ;FINISHED ALL ILF CODES GET OUT
006740  000167  177304                  JMP     MRLF1           ;START NEXT ILF FUNCTION
006744                          ILFDON:
```

```
1919                                    ;*****************************************************************************
1920                                    ;TEST 40                    TEST NO-OP CODES 1 AND 21
1921                                    ;*****************************************************************************
1922  006744  104400                    TST40:  SCOPE
1923
1924                                    ;MODULE TESTED M7759
1925  006746  104414                    MROP:   CLRDK                   ;CLEAR ALL DRIVE REGISTERS
1926  006750  042767  000004  172212            BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
1927  006756  104416                            MRDMD                   ;PUT DRIVE INTO MAINT MODE
1928  006760  104420                            MRCK                    ;CHECK MR REG TO
1929  006762  022701                            2270:                   ;EQUAL 22701
1930  006764  104424                            MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
1931  006766  032767  000010  172174            BIT     #BIT3,ONCEE     ;TESTING CODE 1
1932  006774  001031                            BNE     3$              ;NO CODE 21
1933  006776  012777  000001  172074            MOV     #1,@RSCS1       ;LOAD NO-OP FUNCTION
1934  007004  012767  000001  172204            MOV     #1,WORK         ;LOAD NO-OP FUNCTION
1935  007012  005777  172076                    TST     @RSER           ;ANY ERRORS
1936  007016  001403                            BEQ     1$              ;NO
1937  007020  004767  012114                    JSR     PC,NOPERR       ;TYPE IT
1938  007024  104040                            HLT     !D$             ;TYPE ERROR
1939  007026  022777  010600  172056  1$:       CMP     #10600,@RSDS    ;IS RSDS CORRECT
1940  007034  001403                            BEQ     2$              ;YES
1941  007036  004767  012076                    JSR     PC,NOPERR       ;RSDS SHOULD
1942  007042  104040                            HLT     !D$             ;EQUAL 10600
1943  007044  042767  000004  172116  2$:       BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
1944
1945                                    ;TEST NO-OP FUNCTION CODE 21
1946
1947  007052  052767  000010  172110            BIS     #BIT3,ONCEE     ;TEST TESTING CODE 21 FLAG
1948  007060  012767  000021  172130  3$:       MOV     #21,WORK        ;LOAD CODE 21
1949  007066  012777  000021  172004            MOV     #21,@RSCS1      ;LOAD FUNCTION
1950  007074  005777  172014                    TST     @RSER           ;ANY ERRORS?
1951  007100  001403                            BEQ     4$              ;NO
1952  007102  004767  012032                    JSR     PC,NOPERR       ;YES, TYPE ERROR
1953  007106  104040                            HLT     !D$             ;ERROR DURING NO-OP FUNCTION
1954  007110  022777  010600  171774  4$:       CMP     #10600,@RSDS    ;IS RSDS CORRECT
1955  007116  001403                            BEQ     5$              ;YES
1956  007120  004767  012014                    JSR     PC,NOPERR       ;TYPE ERROR
1957  007124  104040                            HLT     !D$             ;RSDS SHOULD=10600
1958  007126  042767  000014  172034  5$:       BIC     #14,ONCEE       ;CLEAR TEST BITS
```

```
                                ;*******************************************************************************
                                ;TEST 41                    TEST NO-OP FUNCTION WITH ERROR BITS SET
                                ;*******************************************************************************
007134  104400                  TST41:  SCOPE

                                ;MODULE TESTED M7759
007136  104414                  MROPER: CLRDK                   ;CLEAR ALL REGISTERS
007140  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
007142  104420                          MRCK                    ;CHECK MR REG
007144  022701                          22701                   ;TO EQUAL 22701
007146  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
007150  104430                          MRIND                   ;SEND INDEX PULSE

007152  012777  177777  171734          MOV     #-1,@RSER       ;LOAD RSER WITH ERRORS
007160  116701  172002                  MOVB    UNCMP,GOOD      ;GET DRIVE UNDER TEST
007164  017700  171726                  MOV     @RSAS,BAD       ;GET RSAS REG
007170  020100                          CMP     GOOD,BAD        ;DID ATA BIT SET CAUSED BY ERROR
007172  001427                          BEQ     1$              ;YES
007174  104402  007200                  TYPE    ,.+2            ;.ASCIZ <15><12>"SET ERRORS IN RSER-RSAS IS INCORRECT"
007250  104000                          HLT                     ;RSAS=BAD GOOD=CORRECT ANS
007252  012767  000001  171736  1$:     MOV     #1,WORK         ;SETUP FOR NO-OP CODE 1
007260  032767  000010  171702          BIT     #BIT3,ONCEE     ;TESTING CODE 21?
007266  001004                          BNE     2$              ;YES
007270  012777  000001  171602          MOV     #1,@RSCS1       ;SEND NO-OP CODE 1
007276  000406                          BR      3$              ;CHECK FOR ERRORS
007300  012767  000021  171710  2$:     MOV     #21,WORK        ;SETUP FOR CODE 21
007306  012777  000021  171564          MOV     #21,@RSCS1      ;SENT NO-OP CODE 21
007314  017700  171574          3$:     MOV     @RSER,BAD       ;GET RSER REG
007320  012701  177017                  MOV     #177017,GOOD    ;GET CORRECT ANS
007324  020100                          CMP     GOOD,BAD        ;DID RSER CHANGE WITH NO-OP
007326  001411                          BEQ     4$              ;NO
007330  104402  007334                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSER "
007344  004767  011664                  JSR     PC,CHG
007350  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
007352  017700  171540          4$:     MOV     @RSAS,BAD       ;GET RSAS
007356  116701  171604                  MOVB    UNCMP,GOOD      ;GET CORRECT ANS
007362  020100                          CMP     GOOD,BAD        ;IS RSAS CORRECT
007364  001411                          BEQ     5$              ;YES
007366  104402  007372                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSAS "
007402  004767  011626                  JSR     PC,CHG          ;TYPE ERROR
007406  104000                          HLT                     ;RSAS=BAD GOOD=CORRECT ANS
007410  017700  171476          5$:     MOV     @RSDS,BAD       ;GET RSDS
007414  012701  150600                  MOV     #150600,GOOD    ;GET CORRECT ANS
007420  020100                          CMP     GOOD,BAD        ;DID RSDS CHANGE
007422  001411                          BEQ     6$              ;NO
007424  104402  007430                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSDS "
007440  004767  011570                  JSR     PC,CHG          ;TYPE ERROR
007444  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
007446  032767  000010  171514  6$:     BIT     #BIT3,ONCEE     ;TESTING CODE 21
007454  001005                          BNE     7$              ;YES, GET OUT
007456  052767  000010  171504          BIS     #BIT3,ONCEE     ;SET CODE 21 FLAG
007464  000167  177446                  JMP     MROPER          ;TEST CODE 21
007470  042767  000010  171472  7$:     BIC     #BIT3,ONCEE     ;DONE CLEAR FLAG AND CONT.
```

```
                        ;*****************************************************************************
                        ;TEST 42                    BLOCK SEARCH TEST 1
                        ;*****************************************************************************
007476  104400           TST42:  SCOPE

                        ;MODULE TESTED: M7759, M7754, M7771, M7770
                        ;A DRIVE SEARCH FUNCTION IS GIVEN TO THE DRIVE FOR SECTOR 3.
                        ;(SECTOR 41, IF SECTOR INTERLEAVING IS ENABLED) THE
                        ;POSITIONING IN PROGRESS BIT (PIP) AND THE DRIVE READY BIT
                        ;(DRY) IN THE DRIVE STATUS REGISTER (RSDS) ARE CHECKED. THE
                        ;ADDRESS CONFIRM BIT (AC) IS ALSO CHECKED.

007500  104414           MRSRCH: CLRDK                   ;CLEAR ALL REGISTERS
007502  052767  000040  171460        BIS     #BIT5,ONCEE     ;SET CLOCK FLAG
007510  104416                      MRDMD                   ;PUT DRIVE INTO MAINTENANCE MOE
007512  104420                      MRCK                    ;CHECK MR REG
007514  022701                      22701                   ;TO EQUAL 22701
007516  104424                      MRINT                   ;INIT MR REG (CLEAR MRSP)
007520  104430                      MRIND                   ;CLOCK INDEX PULSE IN RSMR
007522  012777  000003  171360      MOV     #3,@RSDA        ;DO A SEARCH FOR SECTOR 3 OR 41
007530  022777  000000  171370      CMP     #0,@RSDT        ;INTERLEAVED?
007536  001403                      BEQ     4$              ;NO SECTOR 3
007540  012777  000041  171342      MOV     #41,@RSDA       ;YES SECTOR 41
007546  012777  000031  171324  4$: MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759)
007554  104426                      DSCK                    ;CHECK RSDS
007556  030400                      30400                   ;TO EQUAL 30400
007560  104000                      HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                            ;BE 0 FOR A DRIVE SEARCH CMD
007562  012767  010643  171414      MOV     #10643,REPT     ;STEP THROUGH 3 SECTORS
007570  104422              1$:     MRCLK                   ;CLOCK MR
007572  104426                      DSCK                    ;RSDS SHOULD NOT
007574  030400                      30400                   ;CHANGE TILL CLOCKING IS COMPLETED
007576  104000                      HLT                     ;TO REACH SECTOR 3
007600  005367  171400              DEC     REPT            ;KEEP CLOCKING TILL
007604  001371                      BNE     1$              ;SECTOR 3 HAS BEEN REACHED
                        ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
007606  104446                      MCLK1                   ;CLOCK MR REG
007610  104426                      DSCK                    ;CHECK FOR "ATA" AND "DRY"
007612  110600                      110600                  ;TO BE SET IN RSDS FOR
007614  104000                      HLT                     ;SEARCH FUNCTION SHOULD BE COMPLETED
007616  022777  104230  171254      CMP     #104230,@RSCS1  ;SET RSCS1
007624  001401                      BEQ     2$              ;SC IN RSCS1 SHOULD SET BECAUSE OF
007626  104140                      HLT     !DS!AS          ;COMPLETED SEARCH FUNCTION
007630  016777  171330  171260  2$: MOV     UNITSV,@RSAS    ;CLEAR ATA
007636  005777  171254              TST     @RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
007642  001401                      BEQ     3$              ;YES
007644  104140                      HLT     !DS!AS          ;RSAS SHOULD=0
007646  022777  004230  171224  3$: CMP     #4230,@RSCS1    ;DID SC CLEAR BY CLEARING
007654  001401                      BEQ     .+4             ;"ATA" YES
007656  104140                      HLT     !DS!AS          ;NO
```

```
                        ;*****************************************************************************
                        ;TEST 43                    BLOCK SEARCH TEST 2
                        ;*****************************************************************************
007660  104400          TST43:  SCOPE

                        ;MODULE TESTED: M7759, M7754, M7771, M7770
                        ;THIS TEST INITIALIZES A BLOCK SEARCH FUNCTION FOR SECTOR 0, WHEN THE DRIVE
                        ;IS CURRENTLY AT THE DESIRED SECTOR. THE BLOCK SEARCH FUNCTION
                        ;SHOULD NOT BE COMPLETED UNTIL THE DRIVE MAKES A COMPLETE REVOLUTION
                        ;AND REACHES THE BEGINNING OF THE DESIRED SECTOR.

007662  104414          MRSRC:  CLRDK                   ;CLEAR ALL REGISTERS
007664  052767  000040  171276        BIS     #BIT5,ONCEE     ;SET CLOCK FLAG
007672  104416                        MRDMD                   ;PUT DRIVE INTO MAINTENANCE MOE
007674  104420                        MRCK                    ;CHECK MR REG
007676  022701                        22701                   ;TO EQUAL 22701
007700  104424                        MRINT                   ;INIT MR REG (CLEAR MRSP)
                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
007702  104430                        MRIND
007704  104420                        MRCK                    ;CHECK MR REG TO EQUAL
007706  022701                        22701                   ;22701
007710  104000                        HLT
                        ;STEP THRU RESYNC PERIOD
007712  012767  001000  171264        MOV     #512.,REPT
007720  052767  000040  171242        BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
007726  104446          MRRT1:  MCLK1                   ;CLOCK MR REG
007730  104420                  MRCK                    ;CHECK FOR
007732  032711                  32711                   ;CORRECT DATA
007734  104000                  HLT                     ;MR = BAD GOOD = CORRECT DATA
007736  104450                  MCLK0                   ;CLOCK MR REG
007740  104420                  MRCK                    ;CHECK FOR
007742  022701                  22701                   ;CORRECT DATA
007744  104000                  HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
007746  005367  171232          DEC     REPT            ;FINISH LOOPING
007752  001365                  BNE     MRRT1           ;THROUGH RESYNC PERIOD
                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE SP = 0
007754  104446                  MCLK1                   ;CLOCK MR REG
007756  104420                  MRCK                    ;MR SHOULD
007760  032311                  32311                   ;EQUALS 32311
007762  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
007764  104450                  MCLK0                   ;CLOCK MR REG
007766  104420                  MRCK                    ;CHECK MR
007770  022301                  22301                   ;TO EQUAL 22301
007772  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
007774  012767  000100  171202        MOV     #100,REPT       ;STEP INTO SECTOR 0
010002  104422          2$:     MRCLK                   ;CLOCK MR REG
010004  005367  171174          DEC     REPT            ;DO 100 TIMES
010010  001374                  BNE     2$              ;DONE YET? NO BR
010012  012777  000031  171060  4$:     MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759) FOR SECTOR 0
010020  104426                  DSCK                    ;CHECK RSDS
010022  030400                  30400                   ;TO EQUAL 30400
010024  104000                  HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                        ;BE 0 FOR A DRIVE SEARCH CMD
010026  012767  021506  171150        MOV     #21506,REPT     ;STEP 3 SECTORS BEYOND SECTOR 0
```

```
2116  010034  104422                  1$:     MRCLK                   ;CLOCK MR
2117  010036  104426                          DSCK                    ;RSDS SHOULD NOT
2118  010040  030400                          30400                   ;CHANGE TILL CLOCKING IS COMPLETED
2119  010042  104000                          HLT                     ;TO REACH SECTOR 3
2120  010044  005367  171134                  DEC     REPT            ;KEEP CLOCKING TILL
2121  010050  001371                          BNE     1$              ;SECTOR 3 HAS BEEN REACHED
2122                                  ;ASSERT INDEX PULSE TO SIMULATE THE BEGINNING OF THE NEXT REVOLUTION
2123  010052  104430                          MRIND
2124  010054  104420                          MRCK                    ;CHECK MR REG TO EQUAL
2125  010056  022701                          22701                   ;22701
2126  010060  104000                          HLT
2127
2128                                  ;STEP THRU RESYNC PERIOD
2129
2130  010062  012767  001000  171114          MOV     #512.,REPT
2131  010070  052767  000040  171072          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2132  010076  104446          MRWR1:  MCLK1                           ;CLOCK MR REG
2133  010100  104420                  MRCK                            ;CHECK FOR
2134  010102  032711                  32711                           ;CORRECT DATA
2135  010104  104000                  HLT                             ;MR = BAD GOOD = CORRECT DATA
2136  010106  104450                  MCLK0                           ;CLOCK MR REG
2137  010110  104420                  MRCK                            ;CHECK FOR
2138  010112  022701                  22701                           ;CORRECT DATA
2139  010114  104000                  HLT                             ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2140  010116  005367  171062          DEC     REPT                    ;FINISH LOOPING
2141  010122  001365                  BNE     MRWR1                   ;THROUGH RESYNC PERIOD
2142
2143                                  ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2144                                  ;SP=0 EQUALS SECTOR PULSE
2145  010124  104446                  MCLK1                           ;CLOCK MR REG
2146  010126  104420                  MRCK                            ;MR SHOULD
2147  010130  032311                  32311                           ;EQUAL 32311
2148  010132  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
2149  010134  104450                  MCLK0                           ;CLOCK MR REG
2150  010136  104420                  MRCK                            ;CHECK MR
2151  010140  022301                  22301                           ;TO EQUAL 22301
2152  010142  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
2153
2154                                  ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2155  010144  104446                  MCLK1                           ;CLOCK MR REG
2156  010146  104426                  DSCK                            ;CHECK FOR "ATA" AND "DRY"
2157  010150  110600                  110600                          ;TO BE SET IN RSDS FOR
2158  010152  104000                  HLT                             ;SEARCH FUNCTION SHOULD BE COMPLETED
2159  010154  022777  104230  170716          CMP     #104230,@RSCS1  ;SET RSCS1
2160  010162  001401                          BEQ     2$              ;SC IN RSCS1 SHOULD SET BECAUSE OF
2161  010164  104140                          HLT     !DS!AS          ;COMPLETED SEARCH FUNCTION
2162  010166  016777  170772  170722  2$:     MOV     UNITSV,@RSAS    ;CLEAR ATA
2163  010174  005777  170716                  TST     @RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
2164  010200  001401                          BEQ     3$              ;YES
2165  010202  104140                          HLT     !DS!AS          ;RSAS SHOULD=0
2166  010204  022777  004230  170666  3$:     CMP     #4230,@RSCS1    ;DID SC CLEAR BY CLEARING
2167  010212  001401                          BEQ     .+4             ;"ATA" YES
2168  010214  104140                          HLT     !DS!AS  ;NO
```

```
                                        ;*************************************************************************
                                        ;TEST 44                DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)
                                        ;*************************************************************************
010216  104400                          TST44:  SCOPE

                                        ;MODULE TESTED M7759, M7755, M7770
                                        ;RMR ERROR IS CAUSED BY WRITTING INTO RSCS1 WHILE DOING A BLOCK SEARCH FUNCTION
                                        ;CHECK RMR DECODER, E12, M7755, IF THIS TEST FAILS

010220  104414                  RMRC1:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
010222  042767  000040  170740          BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
010230  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
010232  104420                          MRCK                    ;CHECK MR REG TO
010234  022701                          22701                   ;EQUAL 22701
010236  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
010240  012777  000001  170642          MOV     #1,@RSDA        ;LOAD RSDA
010246  012777  000031  170624          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
010254  104426                          DSCK                    ;CHECK RSDS
010256  030400                          30400                   ;TO EQUAL 30400
010260  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
010262  012777  000011  170610          MOV     #11,@RSCS1      ;LOAD A CLEAR FUNCTION
                                                                ;THIS SHOULD CAUSE AN RMR
                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                ;WHEN CLEAR COMMAND WAS GIVEN
010270  017700  170620                  MOV     @RSER,BAD       ;GET RSER REG
010274  012701  000004                  MOV     #4,GOOD         ;GET CORRECT ANS
010300  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
010302  001410                          BEQ     1$              ;YES
010304  104402  021305                  TYPE    ,TRMR
010310  104402  010314                  TYPE    ,.+2            ;.ASCIZ "RSCS1"
010322  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
010324  104426                  1$:     DSCK                    ;CHECK RSDS TO
010326  150600                          150600                  ;EQUAL 150600
010330  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
010332  022777  104230  170540          CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
010340  001401                          BEQ     2$              ;YES
010342  104040                          HLT     !DS             ;RSCS1 SHOULD=104230
                                                                ;RSDS SHOULD=150600
                                                                ;RSER SHOULD=4
010344  022777  000001  170536  2$:     CMP     #1,@RSDA        ;DID CLR CLEAR RSDA
010352  001401                          BEQ     4$              ;NO
010354  104004                          HLT     !DA             ;RSDA SHOULD=1
010356  104414                  4$:     CLRDK                   ;CLEAR ALL REGISTERS
010360  005777  170530                  TST     @RSER           ;RSER SHOULD CLEAR
010364  001401                          BEQ     3$              ;RSER OK
010366  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
010370  022777  004200  170502  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
010376  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
010400  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*************************************************************************
                                ;TEST 45                    DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)
                                ;*************************************************************************
010402  104400                  TST45:  SCOPE

                                ;MODULE TESTED M7755 M7759 M7770
                                ;RMR ERROR IS CAUSED BY WRITTING INTO RSDA WHILE DOING A BLOCK SEARCH FUNCTION

010404  104414                  RMRC2:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
010406  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
010410  104420                          MRCK                    ;CHECK MR REG TO
010412  022701                          22701                   ;EQUAL 22701
010414  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
010416  012777  000001  170464          MOV     #1,@RSDA        ;LOAD RSDA
010424  012777  000031  170446          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
010432  104426                          DSCK                    ;CHECK RSDS
010434  030400                          30400                   ;TO EQUAL 30400
010436  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSUED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
010440  005077  170444                  CLR     @RSDA           ;MODIFY RSDA
                                                                ;THIS SHOULD CAUSE AN RMR
                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                ;WHEN COMMAND WAS GIVEN
010444  017700  170444                  MOV     @RSER,BAD       ;GET RSER REG
010450  012701  000004                  MOV     #4,GOOD         ;GET CORRECT ANS
010454  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
010456  001410                          BEQ     1$              ;YES
010460  104402  021305                  TYPE    ,TRMR
010464  104402  010470                  TYPE    ,.+2            ;.ASCIZ "RSDA"
010476  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
010500  104426          1$:             DSCK                    ;CHECK RSDS TO
010502  150600                          150600                  ;EQUAL 150600
010504  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
010506  022777  104230  170364          CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
010514  001401                          BEQ     2$              ;YES
010516  104040                          HLT     !DS             ;RSCS1 SHOULD=104230
                                                                ;RSDS SHOULD=150600
                                                                ;RSER SHOULD=4
010520  022777  000001  170362  2$:     CMP     #1,@RSDA        ;DID CLR CLEAR RSDA
010526  001401                          BEQ     4$              ;NO
010530  104004                          HLT     !DA             ;RSDA SHOULD=1
010532  104414          4$:             CLRDK                   ;CLEAR ALL REGISTERS
010534  005777  170354                  TST     @RSER           ;RSER SHOULD CLEAR
010540  001401                          BEQ     3$              ;RSER OK
010542  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
010544  022777  004200  170326  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
010552  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
010554  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*****************************************************************************
                                ;TEST 46                    DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)
                                ;*****************************************************************************
010556  104400                  TST46:  SCOPE

                                ;MODULE TESTED M7759, M7755, M7770
                                ;RMR ERROR IS CAUSED BY WRITTING INTO RSER WHILE DOING A BLOCK SEARCH FUNCTION
                                ;CHECK RMR DECODER, E12-M7755,IF THIS TEST FAILS.

010560  104414                  RMRC3:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
010562  042767  000040  170400          BIC     #BIT5,ONCEE     ;CLEAR CLOCK COUNT FLAG
010570  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
010572  104420                          MRCK                    ;CHECK MR REG TO
010574  022701                          22701                   ;EQUAL 22701
010576  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
010600  012777  000001  170302          MOV     #1,@RSDA        ;LOAD RSDA
010606  012777  000031  170264          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
010614  104426                          DSCK                    ;CHECK RSDS
010616  030400                          30400                   ;TO EQUAL 30400
010620  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
010622  012777  177777  170264          MOV     #-1,@RSER       ;MODIFY RSER
                                                                ;THIS SHOULD CAUSE AN RMR
                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                ;WHEN COMMAND WAS GIVEN
010630  017700  170260                  MOV     @RSER,BAD       ;GET RSER REG
010634  012701  000004                  MOV     #4,GOOD         ;GET CORRECT ANS
010640  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
010642  001410                          BEQ     1$              ;YES
010644  104402  021305                  TYPE    ,TRMR
010650  104402  010654                  TYPE    ,.+2            ;.ASCIZ "RSER"
010662  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
010664  104426          1$:     DSCK                            ;CHECK RSDS TO
010666  150600                  150600                          ;EQUAL 150600
010670  104000                  HLT                             ;RSDS=BAD GOOD=CORRECT ANS
010672  022777  104230  170200          CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
010700  001401                          BEQ     4$              ;YES
010702  104040                          HLT     !DS             ;RSCS1 SHOULD=104230
                                                                ;RSDS SHOULD=150600
                                                                ;RSER SHOULD=4
010704  104414          4$:     CLRDK                           ;CLEAR ALL REGISTERS
010706  005777  170202                  TST     @RSER           ;RSER SHOULD CLEAR
010712  001401                          BEQ     3$              ;RSER OK
010714  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
010716  022777  004200  170154  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
010724  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADER IN RSCS2
010726  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*************************************************************************
                                ;TEST 47                 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)
                                ;*************************************************************************
010730  104400                  TST47:  SCOPE

                                ;MODULE TESTED: M7759, M7755, M7770
                                ;RMR ERROR SHOULD NOT SET BY WRITTING INTO RSAS WHILE DOING A BLOCK SEARCH FUNCTION
                                ;IF TEST FAILS, CHECK RMR DECODER E12-M7755.

010732  104414                  RMRC4:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
010734  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
010736  104420                          MRCK                    ;CHECK MR REG TO
010740  022701                          22701                   ;EQUAL 22701
010742  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
010744  012777  000001  170136          MOV     #1,@RSDA        ;LOAD RSDA
010752  012777  000031  170120          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
010760  104426                          DSCK                    ;CHECK RSDS
010762  030400                          30400                   ;TO EQUAL 30400
010764  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
010766  005077  170124                  CLR     @RSAS           ;WRITE INTO ATTENTION SUMMARY REGISTER
                                                                ;SHOULD BE NO RMR ERROR BECAUSE
                                                                ;WRITING RSAS IN ALLOWED ANYTIME.
010772  017700  170116                  MOV     @RSER,BAD       ;GET RSER REG
010776  012701  000000                  MOV     #0,GOOD         ;GET CORRECT ANS
011002  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
011004  001435                          BEQ     1$              ;NO
011006  104402  011012                  TYPE    ,.+2            ;.ASCIZ <15><12>"RMR ERROR SHOULD NOT SET WHILE WRITING
011076  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
011100  104426                  1$:     DSCK                    ;CHECK RSDS TO
011102  030400                          30400                   ;EQUAL 30400
011104  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
011106  022777  004231  167764          CMP     #4231,@RSCS1    ;DID CORRECT BITS SET IN RSCS1
011114  001401                          BEQ     4$              ;YES
011116  104040                          HLT     !DS             ;RSCS1 SHOULD=4231
                                                                ;RSDS SHOULD=30400
                                                                ;RSER SHOULD=0
011120  104414                  4$:     CLRDK                   ;CLEAR ALL REGISTEDS
011122  005777  167766                  TST     @RSER           ;RSER SHOULD CLEAR
011126  001401                          BEQ     3$              ;RSER OK
011130  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
011132  022777  004200  167740  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
011140  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADER IN RSCS2
011142  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
2371                                    ;*****************************************************************************
2372                                    ;TEST 50                    DRIVE SELECT TEST
2373                                    ;*****************************************************************************
2374  011144  104400                    TST50:  SCOPE
2375
2376                                    ;MODULE TESTED: M7755
2377                                    ;THE PROGRAM LOADS A DRIVE REGISTER, OF THE DRIVE UNDER TEST, TO ALL ONES.
2378                                    ;THE PROGRAM THEN FINDS A NON-EXISTENT DRIVE AND TRIES TO LOAD ITS
2379                                    ;REGISTER WITH ALL ZEROS. THIS SHOULD CAUSE "NED" TO
2380                                    ;SET IN RSCS2. THE PROGRAM RE-SELECTS THE DRIVE UNDER TEST AND CHECKS
2381                                    ;ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.
2382                                    ;CHECK UNIT NO. COMPARATOR, E19-M7755 IF TEST FAILS
2383
2384  011146  104414                    MRDSEL: CLRDK                   ;CLEAR ALL REGISTERS
2385  011150  104416                            MRDMD                   ;PUT DRIVE INTO MAINT MODE
2386  011152  104420                            MRCK                    ;CHECK MAINT REG
2387  011154  022701                            22701                   ;TO EQUAL 22701
2388  011156  104424                            MRINT                   ;INITIALIZE MAINT MODE (CLEAR MRSP)
2389                                                                    ;BY SENDING 2 CLOCK PULSES
2390  011160  012777  177777  167722            MOV     #-1,@RSDA       ;LOAD DISK ADDR REG OF DRIVE UNDER TEST
2391
2392                                    ;SEARCH FOR NON EXISTENT DRIVES
2393
2394  011166  012767  000401  170022            MOV     #401,WORK
2395  011174  005001                            CLR     GOOD
2396  011176  010177  167700            1$:     MOV     GOOD,@RSCS2     ;LOAD UNIT NO
2397  011202  005777  167706                    TST     @RSER           ;IS THIS A NED?
2398  011206  032777  010000  167666            BIT     #BIT12,@RSCS2   ;IS THIS A NED?
2399  011214  001005                            BNE     2$              ;FOUND NED
2400  011216  005201                            INC     GOOD            ;UPDATE UNIT NUMBER
2401  011220  006167  167772                    ROL     WORK            ;KEEP LOOKING FOR NED
2402  011224  103460                            BCS     NEDDON          ;COULD NOT FIND ANY NON EXISTENT DRIVES
2403  011226  000763                            BR      1$              ;LOOK FOR NED
2404  011230  012777  004000  167642  2$:       MOV     #4000,@RSCS1    ;CLEAR NED
2405  011236  010167  167760                    MOV     GOOD,WORK1      ;SAVE NED NUMBER
2406  011242  010177  167634                    MOV     GOOD,@RSCS2     ;LOAD UNIT # OF NED INTO RSCS2
2407  011246  005077  167636                    CLR     @RSDA           ;WRITE INTO A NON EXISTENT DRIVE REG
2408                                                                    ;THIS SHOULD CAUSE NED TO
2409                                                                    ;SET IN RSCS2
2410  011252  017700  167624                    MOV     @RSCS2,BAD      ;GET RSCS2
2411  011256  052701  010100                    BIS     #10100,GOOD     ;PUT CORRECT ANS IN GOOD
2412                                                                    ;BY SETTING NED AND IR
2413  011262  020100                            CMP     GOOD,BAD        ;IS RSCS2 CORRECT?
2414  011264  001401                            BEQ     .+4             ;YES
2415  011266  104000                            HLT                     ;RSCS2=BAD GOOD=CORRECT ANS
2416
2417  011270  022777  160200  167602            CMP     #160200,@RSCS1  ;IS CS1 CORRECT
2418  011276  001401                            BEQ     .+4             ;YES
2419  011300  104004                            HLT     !DA             ;TRE SHOULD BE SET IN CS1 BECAUSE
2420                                                                    ;OF NED ERROR IN RSCS2
2421                                                                    ;RSCS1 SHOULD=160200
```

```
2422  011302  005777  167610              TST     @RSAS           ;DID ANY ATTENTION BITS SET?
2423  011306  001401                      BEQ     .+4             ;NO
2424  011310  104100                      HLT     !AS             ;NO ATTENTION BITS SHOULD BE SET
2425  011312  112777  000100  167614      MOVB    #100,@RSCS1B    ;CLEAR TRE
2426  011320  032777  010000  167554      BIT     #NED,@RSCS2     ;DID NED CLEAR
2427  011326  001401                      BEQ     .+4             ;YES
2428  011330  104040                      HLT     !DS             ;NED DID NOT CLEAR IN RSCS2
2429                                                              ;BY CLEARING TRE BIT IN RSCS1
2430  011332  016777  167624  167542      MOV     UNNUM,@RSCS2    ;LOAD CORRECT UNIT NUMBER
2431  011340  022777  177777  167542      CMP     #-1,@RSDA       ;DID RSDA GET MODIFIED
2432                                                              ;WHILE WRITING INTO A NON
2433                                                              ;EXISTENT DRIVE?
2434  011346  001443                      BEQ     NNDD            ;NO
2435  011350  104004                      HLT     !DA             ;RSDA SHOULD= -1
2436  011352  016700  167644              MOV     WORK1,BAD       ;IT GOT MODIFIED WHILE WRITING
2437  011356  016701  167600              MOV     UNNUM,GOOD      ;INTO A NED
2438  011362  104000                      HLT                     ;GOOD=DRIVE UNDER TEST
2439  011364  000434                      BR      NNDD            ;BAD=NON EXISTENT DRIVE THAT WAS
2440                                                              ;IN RSCS2 WHEN RSDA GOT MODIFIED
2441  011366  032767  010000  167574  NEDDON: BIT     #BIT12,ONCEE    ;WAS THIS TYPED BEFORE?
2442  011374  001030                      BNE     NNDD            ;YES
2443  011376  104402  011402              TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
2444  011450  052767  010000  167512      BIS     #BIT12,ONCEE    ;SET TYPED MESSAGE FLAG
2445  011456                      NNDD:
```

```
                        ;*****************************************************************************
                        ;TEST 51                    MAINTENANCE MODE WRITE TEST
                        ;*****************************************************************************
011456  104400          TST51:  SCOPE

                        ;MODULE TESTED: M7771, M7753, M7751
                        ;THIS IS AN RS03 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR
                        ;WRITE TEST. WE ARE TESTING THE COMPLETE DATA PATH FOR A DATA
                        ;TRANSFER TO THE DISK. MILLER ENCODED DATA TO BOTH SURFACES IS
                        ;CHECKED ALONG WITH CORRECT GENERATION OF THE CRC WORD AT THE END
                        ;OF THE SECTOR. INDEX PULSES, RESYNC, TIMING PREAMBLE, AND SECTOR
                        ;PULSES ARE ALSO CHECKED.

011460  012767  001602  167456          MOV     #1602,FLAG2     ;SET TEST FLAG
011466  104414  MRWRT:  CLRDK           ;CLEAR DRIVE REGISTERS
011470  012767  000040  167472          MOV     #40,ONCEE       ;SETUP TEST FLAGS
011476  104430          MRIND           ;SEND INDEX PULSE TO MR REG
011500  104420          MRCK            ;CHECK MR REG
011502  022701          22701           ;TO EQUAL 22701
011504  104424          MRINT           ;INIT MAINT MODE (CLEAR MRSP)
                                        ;BY SENDING 2 CLOCK PULSES

                        ;FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
                        ;DATA BUFFER WORDS ARE  :A WORD OF ALL 0'S
                        ;                       A WORD OF ALL 1'S
                        ;                       FLOATING 1'S PATTERN (16 WORDS)
                        ;                       A PATTERN OF 146314 (46 WORDS)

011506  012702  026666          MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
011512  005022          CLR     (R2)+           ;CLEAR 1ST LOCATION
011514  012722  177777          MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
011520  005003          CLR     R3              ;CLEAR WORK LOC TO GENERATE
011522  000261          SEC                     ;A PATTERN OF FLOATING ONES
011524  006103  1$:     ROL     R3              ;GET PATTERN
011526  103402          BCS     2$              ;DONE GET OUT
011530  010322          MOV     R3,(R2)+        ;FILL BUFFER
011532  000774          BR      1$              ;CONT
011534  012703  000056  2$:     MOV     #46.,R3         ;FILL REMAINING PORTION OF

011540  012704  146314          MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
011544  010422  3$:     MOV     R4,(R2)+        ;LOAD BUFFER
011546  005303          DEC     R3              ;DONE YET?
011550  001375          BNE     3$              ;NO

                        ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) TO SECTOR 0

011552  012777  026666  167326          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
011560  012777  177700  167316          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
011566  012777  000061  167304          MOV     #61,@RSCS1      ;LOAD WRITE COMMAND
011574  104454          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
011576  104220          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
011600  104456          SPASS                   ;CLOCK MR REG SP = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
011602  104430                                  MRIND
011604  104420                                  MRCK                    ;CHECK MR REG TO EQUAL
011606  020501                                  20501                   ;20501 FOR A
011610  104000                                  HLT                     ;WRITE COMD HAS BEEN ISSUED

                                        ;STEP THRU RESYNC PERIOD

011612  012767  001000  167364                  MOV     #512.,REPT
011620  052767  000040  167342                  BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
011626  104446          MRWRT1:                 MCLK1                   ;CLOCK MR REG
011630  104420                                  MRCK                    ;CHECK FOR
011632  030511                                  30511                   ;CORRECT DATA
011634  104000                                  HLT                     ;MR = BAD GOOD = CORRECT DATA
011636  104450                                  MCLKO                   ;CLOCK MR REG
011640  104420                                  MRCK                    ;CHECK FOR
011642  020501                                  20501                   ;CORRECT DATA
011644  104000                                  HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
011646  005367  167332                          DEC     REPT            ;FINISH LOOPING
011652  001365                                  BNE     MRWRT1          ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
011654  104446                                  MCLK1                   ;CLOCK MR REG
011656  104420                                  MRCK                    ;MR SHOULD
011660  030111                                  30111                   ;EQUAL 30111
011662  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
011664  104450                                  MCLKO                   ;CLOCK MR REG
011666  104420                                  MRCK                    ;CHECK MR
011670  020101                                  20101                   ;TO EQUAL 20101
011672  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 63 MAINT CLOCK OPERATIONS--WRITING PREAMBLE

011674  012767  000077  167302                  MOV     #63.,REPT
011702  104446          MRWRT2:                 MCLK1                   ;CLOCK MR REG
011704  104420                                  MRCK                    ;CHECK MR REG
011706  031511                                  31511                   ;TO EQUAL 31511
011710  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
011712  104450                                  MCLKO                   ;CLOCK MR REG
011714  104420                                  MRCK                    ;CHECK MR REG
011716  021501                                  21501                   ;TO EQUAL 21501
011720  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
011722  005367  167256                          DEC     REPT            ;DONE YET
011726  001365                                  BNE     MRWRT2          ;NO LOOP
```

```
                                ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
                                        MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        131511                  ;TO EQUAL 131511
                                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                        MCLK0                   ;CLOCK MR REG
                                        MRCK                    ;MR REG SHOULD
                                        25501                   ;EQUAL 25501
                                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                        MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;MR SHOULD EQUAL
                                        135511                  ;35511
                                        HLT
                                ;PERFORM NEXT STEP 8 TIMES TO FINISH WRITING PREAMBLE
000010  167216                          MOV     #10,REPT
                                MRWRT3: MCLK0                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        25501                   ;TO EQUAL 25501
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS
                                        MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        135511                  ;TO EQUAL 135511
                                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
167172                                  DEC     REPT            ;DONE YES?
                                        BNE     MRWRT3          ;NO LOOP BACK

                                ;MOVE DATA WORD INTO RS03 SHIFT REGISTER (M7753)

                                        MCLK0                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        21501                   ;TO EQUAL 21501
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;ENCODE SYNC 1 (M7751)

                                        MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;MR REG SHOULD
                                        123511                  ;EQUAL 123511
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS
                                        MCLK0                   ;CLOCK MR REG
                                        MRCK                    ;MR REG SHOULD NOW
                                        33501                   ;EQUAL 33501
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS
026666                                  MOV     #INBUF,R5       ;GET STARTING ADDR FOR DATA BUFFER
                                        MOV     (R5),R4         ;GET DATA
```

```
2591  012052  012767  002156  167136          MOV     #1134.,WORK     ;DOING A 1 SECTOR TRANSFER 63 WORDS
2592                                                                    ;18 BITS PER WORD-CLOCK LOOPS
2593                                                                    ;TAKE CARE OF 1 BIT AT A TIME
2594                                                                    ;63 TIMES 18 EQUALS 1134 LOOPS
2595                                                                    ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY)
2596  012060  052767  000100  167102          BIS     #BIT6,ONCEE     ;SET 1ST TRANSFER WORD FLAG
2597  012066  104432                  1$:     XBIT                    ;GET 1 BIT OF DATA
2598  012070  104434                          CLKD1                   ;SET MCLK IN RSMR
2599                                                                    ;AND CALCULATE MR REG
2600                                                                    ;FOR CORRECT DATA (MWDB)
2601  012072  104000                          HLT                     ;MR REG NOT CORRECT
2602  012074  104436                          CLKD0                   ;CLEAR MCLK TO
2603                                                                    ;COMPLETE TRANSFER OF THIS BIT
2604                                                                    ;CALCULATE CORRECT ANS FOR
2605                                                                    ;MR REG (MWDB)
2606  012076  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2607  012100  032767  000200  167062          BIT     #BIT7,ONCEE     ;ON LAST WORD YET?
2608  012106  001015                          BNE     2$              ;YES
2609  012110  032767  000400  167052          BIT     #BIT8,ONCEE     ;ON CRC WORD YET?
2610  012116  001040                          BNE     3$              ;YES
2611  012120  005367  167072                  DEC     WORK            ;DONE WITH 63 WORDS?
2612  012124  001360                          BNE     1$              ;NO
2613
2614  012126  052767  000200  167034          BIS     #BIT7,ONCEE     ;SET LAST WORD FLAG
2615  012134  012767  000023  167054          MOV     #19.,WORK       ;SET UP TO TRANSFER LAST WORD
2616  012142  005367  167050          2$:     DEC     WORK            ;DONE YET?
2617  012146  001347                          BNE     1$              ;NO
2618  012150  052767  000400  167012          BIS     #BIT8,ONCEE     ;SET TRANSFERRING CRC WORD
2619  012156  042767  000200  167004          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
2620  012164  004767  011352                  JSR     PC,GENCRC       ;GENERATE CRC WORD
2621                                                                    ;AND LEAVE IN "WORK"
2622  012170  012702  026666                  MOV     #INBUF,R2       ;GO TO END
2623  012174  062702  000200                  ADD     #200,R2         ;OF DATA BUFFER
2624  012200  016712  167012                  MOV     WORK,@R2        ;LOAD CRC WORD
2625  012204  010205                          MOV     R2,R5           ;RESET POINTER FOR
2626  012206  162705  000002                  SUB     #2,R5           ;R5 FOR CRC WD
2627  012212  012767  000023  166776          MOV     #19.,WORK       ;SETUP TO XFER CRC
2628  012220  005367  166772          3$:     DEC     WORK            ;DONE YET
2629  012224  001320                          BNE     1$              ;NO
2630
2631                          ;EBL SHOULD NOW ASSERT
2632
2633  012226  104446                  MCLK1                           ;CLOCK MR REG TO STOP THROUGH
2634                                                                    ;THE RS03 SECTOR DEAD BAND AREA
2635  012230  104420                  MRCK                            ;CHECK MR REG
2636  012232  113511                  113511                          ;TO EQUAL 113511
2637  012234  104000                  HLT                             ;MR REG=BAD GOOD=CORRECT ANS
```

```
                                        ;LOOP 17 TIMES
                                        
012236  012767  000017  166740          MOV     #17,REPT
012244  104450                  4$:     MCLKO                   ;CLOCK MR REG
012246  104420                          MRCK                    ;CHECK MR REG
012250  003501                          3501                    ;TO EQUAL 3501
012252  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012254  104446                          MCLK1                   ;CLOCK MR REG
012256  104420                          MRCK                    ;CHECK MR REG
012260  113511                          113511                  ;TO EQUAL 113511
012262  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012264  005367  166714                  DEC     REPT            ;DONE LOOPING YET?
012270  001365                          BNE     4$              ;NO
                                        
                                        ;FINISH UP
012272  104450                          MCLKO                   ;CLOCK MR REG
012274  104420                          MRCK                    ;CHECK MR REG
012276  003501                          3501                    ;TO EQUAL 3501
012300  104000                          HLT                     ;MR REG=BAD GOOD=CORRECT ANS
012302  104446                          MCLK1                   ;CLOCK MR REG
012304  104420                          MRCK                    ;CHECK MR REG
012306  111511                          111511                  ;TO EQUAL 111511
012310  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012312  104450                          MCLKO                   ;CLOCK MR REG
012314  104420                          MRCK                    ;CHECK MRE REG
012316  001501                          1501                    ;TO EQUAL 1501
012320  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
                                        
                                        ;TRANSFER SHOULD NOW BE COMPLETE
                                        
012322  104446                          MCLK1                   ;CLOCK MR REG
012324  104420                          MRCK                    ;CHECK MR
012326  012711                          12711                   ;REG TO
012330  104008                          HLT                     ;EQUAL 12711
012332  104450                          MCLKO                   ;CLOCK MR REG
012334  104420                          MRCK                    ;CHECK MR REG
012336  002701                          2701                    ;TO
012340  104000                          HLT                     ;EQUAL 2701
                                        
                                        ;NOW TEST CONTROLLER
                                        
012342  005777  166532                  TST     @RSCS1          ;ANY ERRORS?
012346  100001                          BPL     5$              ;NO
012350  104014                          HLT     !DA!WC          ;YES
012352  005777  166526          5$:     TST     @RSWC           ;DID WC GO TO 0
012356  001401                          BEQ     .+4             ;YES
012360  104010                          HLT     !WC             ;WC SHOULD BE = TO 0
012362  022777  000001  166520          CMP     #1,@RSDA        ;DID RSDA INCREMENT TO A 1
012370  001401                          BEQ     .+4             ;YES
012372  104004                          HLT     !DA             ;NO RSDA SHOULD=1
012374  032767  000002  166542          BIT     #BIT1,FLAG2     ;IN MAINT VERIFY TEST?
012402  001002                          BNE     .+6             ;NO
012404  000137  020564                  JMP     @#MRVR2         ;YES, GO TO VERIFY TEST
```

```
                                ;*****************************************************************************
                                ;TEST 52                    MAINTENANCE READ TEST
                                ;*****************************************************************************
012410  104400                  TST52:  SCOPE

                                ;MODULE TESTED: M7771, M7753, M7751
                                ;THIS IS AN RS03 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR READ TIMING
                                ;TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE DISK DECODING LOGIC
                                ;TO CORE MEMORY. (THE PHASE LOCK LOOP IS NOT TESTED)

012412  104414                  MRRD:   CLRDK                   ;CLEAR DRIVE REGISTERS
012414  052767  000040  166546          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
012422  042767  047716  166540          BIC     #47716,ONCEE    ;CLEAR ALL OTHER FLAG BITS
012430  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
012432  104420                          MRCK                    ;CHECK MR REG
012434  022701                          22701                   ;TO EQUAL 22701
012436  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES

012440  005067  166500                  CLR     FLAG2           ;CLEAR FLAG TEST BITS

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
                                ;DATA BUFFER WORDS ARE  :A WORD OF ALL 0'S
                                ;                       A WORD OF ALL 1'S
                                ;                       FLOATING 1'S PATTERN (16 WORDS)
                                ;                       A PATTERN OF 146314 (46 WORDS)

012444  012702  026666                  MOV     #INBUF,R2       ;GET LOCATION OF INBUF
012450  005022                          CLR     (R2)+           ;CLEAR 1ST LOCATIGN
012452  012722  177777                  MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
012456  005003                          CLR     R3              ;CLEAR WORK LOC TO GENERATE
012460  000261                          SEC                     ;A PATTERN OF FLOATING ONES
012462  006103                  1$:     ROL     R3              ;GET PATTERN
012464  103402                          BCS     2$              ;DONE GET OUT
012466  010322                          MOV     R3,(R2)+        ;FILL BUFFER
012470  000774                          BR      1$              ;CONT
012472  012703  000056          2$:     MOV     #46.,R3         ;FILL REMAINING PORTION OF
012476  012704  146314                  MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
012502  010422                  3$:     MOV     R4,(R2)+        ;LOAD BUFFER
012504  005303                          DEC     R3              ;DONE YET
012506  001375                          BNE     3$              ;NO

                                ;NOTE
                                ;INBUF CONTAINS THE TABLE OF DATA WHICH IS "READ"
                                ;VIA THE MRDB BIT IN RSMR.
                                ;OUTBUF IS WHERE THE DATA WORDS FROM THE
                                ;MASSBUS ARE STORED.
```

```
2738                                    ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
2739
2740  012510  012777  027466  166370          MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
2741  012516  012777  177700  166360          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
2742  012524  012777  000071  166346          MOV     #71,@RSCS1      ;LOAD READ COMMAND
2743  012532  012702  000100                  MOV     #100.R2         ;CLEAR THE OUTBUF TABLE SO THAT
2744  012536  012703  027466                  MOV     #OUTBUF,R3      ;WHEN THE READ IS FINISHED, WE CAN
2745  012542  005023                  4$:     CLR     (R3)+           ;COMPARE WHAT WE GOT (OUTBUF)
2746  012544  005302                          DEC     R2              ;WITH WHAT WE EXPECTED (INBUF).
2747  012546  001375                          BNE     4$
2748  012550  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
2749                                                                  ;TO CLEAR OUT COUNTERS AND REGISTERS
2750                                                                  ;THAT OTHERWISE COULD NOT BE CLEARED.
2751  012552  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
2752  012554  104456                          SPASS                   ;CLOCK MR REG SP = 1
2753
2754                                    ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2755  012556  104430                          MRIND
2756  012560  104420                          MRCK                    ;CHECK MR REG TO EQUAL
2757  012562  022601                          22601                   ;22601 FOR A
2758  012564  104000                          HLT                     ;READ COMD
2759
2760                                    ;STEP THRU RESYNC PERIOD
2761
2762  012566  012767  001000  166410          MOV     #512.,REPT
2763  012574  052767  000040  166366          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT
2764  012602  104446                  MRRD1:  MCLK1                   ;CLOCK MR REG
2765  012604  104420                          MRCK                    ;CHECK FOR
2766  012606  032611                          32611                   ;CORRECT DATA
2767  012610  104000                          HLT                     ;MR=BAD GOOD=CORRECT DATA
2768  012612  104450                          MCLK0                   ;CLOCK MR REG
2769  012614  104420                          MRCK                    ;CHECK FOR
2770  012616  022601                          22601                   ;CORRECT DATA
2771  012620  104000                          HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
2772  012622  005367  166356                  DEC     REPT            ;FINISH LOOPING
2773  012626  001365                          BNE     MRRD1           ;THROUGH RESYNC PERIOD
2774
2775                                    ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2776                                    ;SP=0 EQUALS SECTOR PULSE
2777  012630  104446                          MCLK1                   ;CLOCK MR REG
2778  012632  104420                          MRCK                    ;MR SHOULD
2779  012634  032211                          32211                   ;EQUAL 32211
2780  012636  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2781  012640  104450                          MCLK0                   ;CLOCK MR REG
2782  012642  104420                          MRCK                    ;CHECK MR
2783  012644  022201                          22201                   ;TO EQUAL 22201
2784  012646  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;PERFORM 71 MAINT CLOCK OPERATIONS--

012650  012767  000107  166326            MOV     #71.,REPT
012656  104446            MRRD2:  MCLK1                   ;CLOCK MR REG
012660  104420                    MRCK                    ;CHECK MR REG
012662  033611                    33611                   ;TO EQUAL 33611
012664  104000                    HLT                     ;MR=BAD GOOD=CORRECT ANS
012666  104450                    MCLK0                   ;CLOCK MR REG
012670  104420                    MRCK                    ;CHECK MR REG
012672  023601                    23601                   ;TO EQUAL 23601
012674  104000                    HLT                     ;MR=BAD GOOD=CORRECT ANS
012676  005367  166302            DEC     REPT            ;DONE YET
012702  001365                    BNE     MRRD2           ;NO LOOP

                                        ;READ SYNC"1"

012704  012777  000005  166212            MOV     #5,@RSMR
012712  012777  000015  166204            MOV     #15,@RSMR
012720  104420                    MRCK
012722  133615                    133615
012724  104000                    HLT

                                        ;READ DATA
012726  005067  166274            MRD3:   CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
012732  012705  026666                    MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
012736  162705  000002                    SUB     #2,R5
012742  012767  000045  166236            MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
012750  012767  002200  166226            MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;64 WORDS-18X64=1152
                                                                ;1 CLOCK PER 1 BIT OF DATA
012756  104444            1$:     RBIT                    ;GET 1 DATA BIT
012760  104440                    CLKR1                   ;CLOCK MR REG
012762  104000                    HLT                     ;MR NOT CORRECT

012764  104442                    CLKR0                   ;CLOCK MR REG
012766  104000                    HLT                     ;MR REG NOT CORRECT

012770  005367  166210            DEC     REPT            ;DONE WITH COMPLETE TRANSFER
012774  001370                    BNE     1$              ;NO
```

```
2824  013776  032767  000400  166164  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
2825  014004  001030                          BNE     3$              ;YES
2826  014006  052767  000400  166154          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
2827  014014  016767  166166  166152          MOV     REPT1,SAVEE     ;SAVE REPT1
2828  014022  004767  010514                  JSR     PC,GENCRC       ;GENERATE CRC WORD
2829                                                                  ;AND LEAVE IN LOC "WORK"
2830  014026  012702  026666                  MOV     #INBUF,R2
2831  014032  016767  166136  166146          MOV     SAVEE,REPT1     ;RESTORE REPT1
2832  014040  062702  000200                  ADD     #200,R2         ;STORE CRC WORD AT END OF
2833  014044  016712  166146                  MOV     WORK,@R2        ;INBUF TABLE
2834  014050  010205                          MOV     R2,R5
2835  014052  162705  000002                  SUB     #2,R5
2836  014056  012767  000022  166120          MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
2837  014064  000734                          BR      1$              ;TRANSFER CRC WD
2838  014066  104446                  3$:     MCLK1                   ;CLOCK MR REG
2839  014070  104420                          MRCK                    ;CHECK MR REG
2840  014072  117611                          117611                  ;TO EQUAL
2841  014074  104000                          HLT                     ;117611
2842  014076  104450                          MCLKO                   ;CLOCK MR REG
2843  014100  104420                          MRCK                    ;CHECK MR
2844  014102  003601                          3601                    ;TO EQUAL
2845  014104  104000                          HLT                     ;3601
2846  014106  104446                          MCLK1                   ;CLOCK MR REG
2847  014110  104420                          MRCK                    ;CHECK MR
2848  014112  113611                          113611                  ;TO EQUAL
2849  014114  104000                          HLT                     ;113611
2850  014116  104450                          MCLKO                   ;CLOCK MR REG
2851  014120  104420                          MRCK                    ;CHECK MR
2852  014122  003601                          3601                    ;TO EQUAL
2853  014124  104000                          HLT                     ;3601
2854
2855                                  ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
2856                                  ;STEP INTO END OF SECTOR DEAD BAND
2857                                  ;EBL IS NOW ASSERTED
2858
2859  014126  012767  000020  166050  MRD4:   MOV     #20,REPT
2860  014134  104446                  1$:     MCLK1                   ;CLOCK MR REG
2861  014136  104420                          MRCK                    ;CHECK MR REG
2862  014140  113611                          113611                  ;TO EQUAL
2863  014142  104000                          HLT                     ;113611
2864  014144  104450                          MCLKO                   ;CLOCK MR REG
2865  014146  104420                          MRCK                    ;CHECK MR
2866  014150  003601                          3601                    ;REG TO
2867  014152  104000                          HLT                     ;EQUAL 3601
2868  014154  005367  166024                  DEC     REPT            ;DONE YET?
2869  014160  001365                          BNE     1$              ;NO
2870
2871                                  ;PERFORM ONE MAINTENANCE CLOCK OPERATION
2872                                  ;SHOULD GET STROBE BUFFER
2873
2874  014162  104446                          MCLK1                   ;CLOCK MR REG
2875  014164  104420                          MRCK                    ;CHECK MR
2876  014166  117611                          117611                  ;REG TO
2877  014170  104000                          HLT                     ;EQUAL 117611
```

```
                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.
                                        
013172  104450                  MRD5:   MCLKD                   ;CLOCK MR REG
013174  022777  004270  165676          CMP     #4270,@RSCS1    ;ANY ERRORS?
013202  001401                          BEQ     1$              ;NO
013204  104054                          HLT     !DA!DS!WC
013206  005777  165672          1$:     TST     @RSWC           ;DID WC GO TO 0
013212  001401                          BEQ     .+4             ;YES
013214  104010                          HLT     !WC             ;WC REG SHOULD=0
013216  022777  000001  165664          CMP     #1,@RSDA        ;DOES RSAD=1
013224  001401                          BEQ     .+4             ;YES
013226  104004                          HLT     !DA             ;NO RSDA SHOULD=1
                                        
                                        ;COMPARE DATA READ WITH INPUT BUFFER
                                        ;WILL ONLY TYPEOUT 10 ERRORS --- BUT IF SW12 IS SET
                                        ;IT WILL TYPE OUT ALL ERRORS
                                        
013230  012700  026666          MRD6:   MOV     #INBUF,BAD      ;GET STARTING LOC OF EXPECTED DATA
013234  012701  027466                  MOV     #OUTBUF,GOOD    ;GET STARTING LOC OF DATA "READ" FROM DISK
013240  012767  000012  165736          MOV     #12,REPT        ;SET UP ERROR COUNTER
013246  012705  000101                  MOV     #101,R5         ;COMPARE 1 SECTOR
013252  005305                  3$:     DEC     R5              ;DONE WITH SECTOR
013254  001433                          BEQ     2$              ;YES GET OUT
013256  022021                          CMP     (BAD)+,(GOOD)+  ;IS DATA CORRECT?
013260  001774                          BEQ     3$              ;YES
013262  032777  010000  164300          BIT     #BIT12,@SWR     ;TYPE ALL ERRORS?
013270  001003                          BNE     1$              ;YES
013272  005367  165706                  DEC     REPT            ;TYPED OUT 10 ERRORS YET?
013276  001422                          BEQ     2$              ;YES GET OUT
013300  024041                  1$:     CMP     -(BAD),-(GOOD)  ;GET ERROR
013302  104000                          HLT                     ;TYPE OUT ERROR
013304  010067  165706                  MOV     BAD,WORK
013310  104402  013314                  TYPE    ,.+2
013332  016746  165660                  MOV     WORK,-(6)       ;PUT WORK ON STACK
013336  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
013340  022021                          CMP     (BAD)+,(GOOD)+
013342  000743                          BR      3$
013344                          2$:                             ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 53                    MAINTENANCE MODE DATA WRITE CHECK TEST
                                ;*****************************************************************************
013344  104400                  TST53:  SCOPE

                                ;MODULE TESTED: M7771, M7753, M7751
                                ;A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION.
                                ;WITHIN THE RS03, A WRITE CHECK FUNCTION IS IDENTICAL TO A
                                ;READ FUNCTION.

013346  104414                  MRWCK:  CLRDK                           ;CLEAR DRIVE REGISTERS
013350  052767  000040  165612          BIS     #BIT5,ONCEE             ;SET TYPE CLOCK COUNT FLAG
013356  042767  047716  165604          BIC     #47716,ONCEE            ;CLEAR ALL OTHER FLAG BITS
013364  104430                          MRIND                           ;SEND INDEX PULSE TO MR REG
013366  104420                          MRCK                            ;CHECK MR REG
013370  022701                          22701                           ;TO EQUAL 22701
013372  104424                          MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                        ;BY SENDING 2 CLOCK PULSES

013374  012767  000004  165542          MOV     #4,FLAG2                ;SET WC FLAG FOR CLKR1 ROUTINE

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
                                ;DATA BUFFER WORDS ARE  :A WORD OF ALL 0'S
                                ;                       A WORD OF ALL 1'S
                                ;                       FLOATING 1'S PATTERN (16 WORDS)
                                ;                       A PATTERN OF 146314 (46 WORDS)

013402  012702  026666                  MOV     #INBUF,R2               ;GET LOCATION OF INBUF
013406  005022                          CLR     (R2)+                   ;CLEAR 1ST LOCATION
013410  012722  177777                  MOV     #-1,(R2)+               ;2ND WORD OF ALL ONES
013414  005003                          CLR     R3                      ;CLEAR WORK LOC TO GENERATE
013416  000261                          SEC                             ;A PATTERN OF FLOATING ONES
013420  006103                  1$:     ROL     R3                      ;GET PATTERN
013422  103402                          BCS     2$                      ;DONE GET OUT
013424  010322                          MOV     R3,(R2)+                ;FILL BUFFER
013426  000774                          BR      1$                      ;CONT
013430  012703  000056          2$:     MOV     #46.,R3                 ;FILL REMAINING PORTION OF
013434  012704  146314                  MOV     #146314,R4              ;BUFFER WITH A PATTERN OF 146314
013440  010422                  3$:     MOV     R4,(R2)+                ;LOAD BUFFER
013442  005303                          DEC     R3                      ;DONE YET
013444  001375                          BNE     3$                      ;NO

                                ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0

013446  012777  026666  165432          MOV     #INBUF,@RSBA            ;LOAD BUS ADDR REG
013454  012777  177700  165422          MOV     #177700,@RSWC           ;LOAD WORD COUNT REG
013462  012777  000051  165410          MOV     #51,@RSCS1              ;LOAD WRITE CHECK COMMAND
013470  104454                          GETSP                           ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
013472  104220                          HLT     !MR                     ;COULD NOT SET SECTOR PULSE (0)
013474  104456                          SPASS                           ;CLOCK MR REG SP = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
013476  104430                                MRIND
013500  104420                                MRCK                    ;CHECK MR REG TO EQUAL
013502  022701                                22701                   ;22701
013504  104000                                HLT

                                        ;STEP THRU RESYNC PERIOD

013506  012767  001000  165470                MOV     #512.,REPT
013514  052767  000040  165446                BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
013522  104446                        MRWCK1: MCLK1                   ;CLOCK MR REG
013524  104420                                MRCK                    ;CHECK FOR
013526  032711                                32711                   ;CORRECT DATA
013530  104000                                HLT                     ;MR=BAD GOOD=CORRECT DATA
013532  104450                                MCLK0                   ;CLOCK MR REG
013534  104420                                MRCK                    ;CHECK FOR
013536  022701                                22701                   ;CORRECT DATA
013540  104000                                HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
013542  005367  165436                        DEC     REPT            ;FINISH LOOPING
013546  001365                                BNE     MRWCK1          ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
013550  104446                                MCLK1                   ;CLOCK MR REG
013552  104420                                MRCK                    ;MR SHOULD
013554  032311                                32311                   ;EQUAL 32311
013556  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
013560  104450                                MCLK0                   ;CLOCK MR REG
013562  104420                                MRCK                    ;CHECK MR
013564  022301                                22301                   ;TO EQUAL 22301
013566  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--

013570  012767  000107  165406                MOV     #71.,REPT
013576  104446                        MRWCK2: MCLK1                   ;CLOCK MR REG
013600  104420                                MRCK                    ;CHECK MR REG
013602  033711                                33711                   ;TO EQUAL 33711
013604  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
013606  104450                                MCLK0                   ;CLOCK MR REG
013610  104420                                MRCK                    ;CHECK MR REG
013612  023701                                23701                   ;TO EQUAL 23701
013614  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
013616  005367  165362                        DEC     REPT            ;DONE YET
013622  001365                                BNE     MRWCK2          ;NO LOOP
```

```
                                        ;READ SYNC"1"
                                        
013624  012777  000005  165272          MOV     #5,@RSMR
013632  012777  000015  165264          MOV     #15,@RSMR
013640  104420                          MRCK
013642  133715                          133715
013644  104000                          HLT
                                        
                                        ;READ DATA
013646  005067  165354          MRWCK3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
013652  012705  026666                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
013656  162705  000002                  SUB     #2,R5
013662  012767  000045  165316          MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
013670  012767  002200  165306          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;64 WORDS-18X64=1152
                                                                ;1 CLOCK PER 1 BIT OF DATA
013676  104444                  1$:     RBIT                    ;GET 1 DATA BITS
013700  104440                          CLKR1                   ;CLOCK MR REG
013702  104000                          HLT                     ;MR NOT CORRECT
                                        
013704  104442                          CLKR0                   ;CLOCK MR REG
013706  104000                          HLT                     ;MR REG NOT CORRECT
                                        
013710  005367  165270                  DEC     REPT            ;DONE WITH COMPLETE TRANSFER
013714  001370                          BNE     1$              ;NO
013716  032767  000400  165244  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
013724  001030                          BNE     3$              ;YES
013726  052767  000400  165234          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
013734  016767  165246  165232          MOV     REPT1,SAVEE     ;SAVE REPT1
013742  004767  007574                  JSR     PC,GENCRC       ;GENERATE CRC WORD
                                                                ;AND LEAVE IN LOC "WORK"
013746  012702  026666                  MOV     #INBUF,R2       ;
013752  016767  165216  165226          MOV     SAVEE,REPT1     ;RESTORE REPT1
013760  062702  000200                  ADD     #200,R2         ;STORE CRC WORD AT END OF
013764  016712  165226                  MOV     WORK,@R2        ;INBUF TABLE
013770  010205                          MOV     R2,R5
013772  162705  000002                  SUB     #2,R5
013776  012767  000022  165200          MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
014004  000734                          BR      1$              ;TRANSFER CRC WD
```

```
                                3$:     MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        117711                  ;TO EQUAL
                                        HLT                     ;117711
                                        MCLK0                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR
                                        3701                    ;TO EQUAL
                                        HLT                     ;3701
                                        MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR
                                        113711                  ;TO EQUAL
                                        HLT                     ;113711
                                        MCLK0                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR
                                        3701                    ;TO EQUAL
                                        HLT                     ;3701

                                ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
                                ;STEP INTO END OF SECTOR DEAD BAND
                                ;EBL IS NOW ASSERTED

                        MRWCK4: MOV     #20,REPT
                                1$:     MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        113711                  ;TO EQUAL
                                        HLT                     ;113711
                                        MCLK0                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR
                                        3701                    ;REG TO
                                        HLT                     ;EQUAL 3701
                                        DEC     REPT            ;DONE YET?
                                        BNE     1$              ;NO

                                ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                ;SHOULD GET STROBE BUFFER

                                        MCLK1                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR
                                        117711                  ;REG TO
                                        HLT                     ;EQUAL 117711

                                ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                ;SHOULD COMPLETE TRANSFER.

                        MRWCK5: MCLK0                           ;CLOCK MR REG
                                CMP     #4250,@RSCS1            ;ANY ERRORS?
                                BEQ     1$                      ;NO
                                HLT     !DA!DS!WC
                                1$:     TST     @RSWC           ;DID WC GO TO 0
                                        BEQ     .+4             ;YES
                                        HLT     !WC             ;WC REG SHOULD=0
                                        CMP     #1,@RSDA        ;DOES RSDA=1
                                        BEQ     .+4             ;YES
                                        HLT     !DA             ;RSDA SHOULD=1
```

```
                                ;*******************************************************************************
                                ;TEST 54                    MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)
                                ;*******************************************************************************
014150  104400                  TST54:  SCOPE

                                ;MODULES TESTED: M7753)
                                ;THE RS03 DISK IS SET UP TO READ (IN MAINTENANCE MODE) ONE
                                ;SECTOR OF A SPECIALLY CREATED DATA PATTERN WHICH LEAVES ONLY
                                ;ONE BIT SET IN THE CRC REGISTER PRIOR TO CHECKING THE CRC
                                ;WORD. THE CORRESPONDING CRC WORD IS THEN "READ", RESULTING
                                ;IN NO DCK ERROR. THE DATA PATTERN IS THEN MODIFIED (BY
                                ;SHIFTING) AND THE ENTIRE READ SEQUENCE REPEATED UNTIL ALL
                                ;16 BITS IN THE CRC REGISTER HAVE BEEN CHECKED.

014152  012767  000040  164764          MOV     #40,FLAG2       ;CLEAR TST FLAG
014160  104414          MRCRC:  CLRDK                           ;CLEAR DRIVE REGISTERS
014162  052767  000040  165000          BIS     #BIT5,ONCEE     ;TYPE CLOCK COUNT IF ERROR OCCURS
014170  042767  047716  164772          BIC     #47716,ONCEE    ;CLEAR ALL OTHER FLAG BITS
014176  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
014200  104420                          MRCK                    ;CHECK MR REG
014202  022701                          22701                   ;TO EQUAL 22701
014204  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
014206  032767  000020  164730          BIT     #BIT4,FLAG2     ;FIRST TIME THROUGH
014214  001023                          BNE     3$              ;NO
014216  012767  000001  164750          MOV     #1,SAVEE        ;LOAD 1ST CRC WORD

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 1 SECTOR
                                ;CREATE BUFFER WITH 72 WORDS OF 16 BITS WHICH EQUALS THE NO. OF BITS IN 64 18 BITS WORDS
                                ;DATA BUFFER CONTAINS 6 WORDS OF ZEROS
                                :                       A WORD OF 236
                                :                       A WORD OF 140000
                                :                       64 WORDS OF ZEROS
                                ;IN THIS TEST, ALL 18 BITS OF THE RS03 DATA WORD MUST BE
                                ;MANIPULATED.  HENCE A TABLE CONTAINING 1152 BITS (64X18) IS
                                ;REQUIRED INSTEAD OF A TABLE CONTAINING 64 WORDS.
014224  012702  026666                  MOV     #INBUF,R2       ;GET LOCATION OF INBUF
014230  012703  000006                  MOV     #6,R3           ;SETUP COUNTER
014234  005022          1$:     CLR     (R2)+                   ;TO CLEAR THE
014236  005303                  DEC     R3                      ;FIRST 6
014240  001375                  BNE     1$                      ;WORDS
014242  012722  000236                  MOV     #236,(R2)+      ;LOAD A 236
014246  012722  140000                  MOV     #140000,(R2)+   ;LOAD A 140000
014252  012703  000100                  MOV     #64.,R3         ;SETUP COUNTER
014256  005022          2$:     CLR     (R2)+                   ;TO CLEAR THE
014260  005303                  DEC     R3                      ;REMAINING WORDS
014262  001375                  BNE     2$                      ;FOR
```

```
                                        ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0

014264  012777  027466  164614  3$:     MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
014272  012777  177700  164604          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
014300  012777  000071  164572          MOV     #71,@RSCS1      ;LOAD READ COMMAND
014306  012702  000200                  MOV     #200,R2
014312  012703  027466                  MOV     #OUTBUF,R3
014316  052767  000020  164620          BIS     #BIT4,FLAG2     ;NO SET FLAG FOR 1ST TIME THROUGH TEST
014324  005023                  4$:     CLR     (R3)+
014326  005302                          DEC     R2
014330  001375                          BNE     4$
014332  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
014334  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
014336  104456                          SPASS                   ;CLOCK MR REG SP = 1

                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
014340  104430                          MRIND
014342  104420                          MRCK                    ;CHECK MR REG TO EQUAL
014344  022601                          22601                   ;22601 FOR A
014346  104000                          HLT                     ;READ COMD

                                        ;STEP THRU RESYNC PERIOD

014350  012767  001000  164626          MOV     #512.,REPT
014356  052767  000040  164604          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
014364  104446          MRCRC1: MCLK1                           ;CLOCK MR REG
014366  104420                  MRCK                            ;CHECK FOR
014370  032611                  32611                           ;CORRECT DATA
014372  104000                  HLT                             ;MR=BAD GOOD=CORRECT DATA
014374  104450                  MCLK0                           ;CLOCK MR REG
014376  104420                  MRCK                            ;CHECK FOR
014400  022601                  22601                           ;CORRECT DATA
014402  104000                  HLT                             ;ERROR WHILE CLOCKING THROUGH RESYNC
014404  005367  164574          DEC     REPT                    ;FINISH LOOPING
014410  001365                  BNE     MRCRC1                  ;THROUGH RESYNC PERIOD

                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                ;SP=0 EQUALS SECTOR PULSE
014412  104446                  MCLK1                           ;CLOCK MR REG
014414  104420                  MRCK                            ;MR SHOULD
014416  032211                  32211                           ;EQUAL 32211
014420  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
014422  104450                  MCLK0                           ;CLOCK MR REG
014424  104420                  MRCK                            ;CHECK MR
014426  022201                  22201                           ;TO EQUAL 22201
014430  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
```

```
                                ;PERFORM 71 MAINT CLOCK OPERATIONS--

014432  012767  000107  164544          MOV     #71.,REPT
014440  104446          MRCRC2: MCLK1                   ;CLOCK MR REG
014442  104420                  MRCK                    ;CHECK MR REG
014444  033611                  33611                   ;TO EQUAL 33611
014446  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
014450  104450                  MCLK0                   ;CLOCK MR REG
014452  104420                  MRCK                    ;CHECK MR REG
014454  023601                  23601                   ;TO EQUAL 23601
014456  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
014460  005367  164520          DEC     REPT            ;DONE YET
014464  001365                  BNE     MRCRC2          ;NO LOOP

                                ;READ SYNC"1"
014466  012777  000005  164430          MOV     #5,@RSMR
014474  012777  000015  164422          MOV     #15,@RSMR
014502  104420                  MRCK
014504  133615                  133615
014506  104000                  HLT

                                ;READ DATA
014510  005067  164512          MRCRC3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
014514  012705  026666                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
014520  162705  000002                  SUB     #2,R5
014524  012767  000045  164454          MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SD BIT
014532  012767  002200  164444          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;64 WORDS-18X64=1152
                                                                ;1 CLOCK PER 1 BIT OF DATA
014540  104444          1$:     RBIT                    ;GET 1 DATA BITS
014542  104440                  CLKR1                   ;CLOCK MR REG
014544  104000                  HLT                     ;MR NOT CORRECT

014546  104442                  CLKR0                   ;CLOCK MR REG
014550  104000                  HLT                     ;MR REG NOT CORRECT

014552  005367  164426          DEC     REPT            ;DONE WITH COMPLETE TRANSFER
014556  001370                  BNE     1$              ;NO
```

```
3240  014560  032767  000400  164402  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
3241  014566  001020                          BNE     3$              ;YES
3242  014570  052767  000400  164372          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
3243  014576  012702  026666                  MOV     #INBUF,R2       ;MOVE CRC
3244  014602  062702  000220                  ADD     #220,R2         ;WORD TO END OF
3245  014606  016712  164362          4$:     MOV     SAVEE,@R2       ;INBUF TABLE
3246  014612  010205                  5$:     MOV     R2,R5           ;GET CRC WORD
3247  014614  162705  000002                  SUB     #2,R5
3248  014620  012767  000022  164356          MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
3249  014626  000744                          BR      1$              ;TRANSFER CRC WD
3250  014630  104446                  3$:     MCLK1                   ;CLOCK MR REG
3251  014632  104420                          MRCK                    ;CHECK MR REG
3252  014634  117611                          117611                  ;TO EQUAL
3253  014636  104000                          HLT                     ;117611
3254  014640  104450                          MCLKO                   ;CLOCK MR REG
3255  014642  104420                          MRCK                    ;CHECK MR
3256  014644  003601                          3601                    ;TO EQUAL
3257  014646  104000                          HLT                     ;3601
3258  014650  104446                          MCLK1                   ;CLOCK MR REG
3259  014652  104420                          MRCK                    ;CHECK MR
3260  014654  113611                          113611                  ;TO EQUAL
3261  014656  104000                          HLT                     ;11361!
3262  014660  104450                          MCLKO                   ;CLOCK MR REG
3263  014662  104420                          MRCK                    ;CHECK MR
3264  014664  003601                          3601                    ;TO EQUAL
3265  014666  104000                          HLT                     ;3601
```

```
                                        ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
                                        ;STEP INTO END OF SECTOR DEAD BAND
                                        ;EBL IS NOW ASSERTED.
014670  012767  000020  164306  MRCRC4: MOV     #20,REPT
014676  104446                1$:       MCLK1                   ;CLOCK MR REG
014700  104420                          MRCK                    ;CHECK MR REG
014702  113611                          113611                  ;TO EQUAL
014704  104000                          HLT                     ;113611
014706  104450                          MCLK0                   ;CLOCK MR REG
014710  104420                          MRCK                    ;CHECK MR
014712  003601                          3601                    ;REG TO
014714  104000                          HLT                     ;EQUAL 3601
014716  005367  164262                  DEC     REPT            ;DONE YET?
014722  001365                          BNE     1$              ;NO

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD GET STROBE BUFFER

014724  104446                          MCLK1                   ;CLOCK MR REG
014726  104420                          MRCK                    ;CHECK MR
014730  117611                          117611                  ;REG TO
014732  104000                          HLT                     ;EQUAL 117611

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.

014734  104450                MRCRC5:   MCLK0                   ;CLOCK MR REG
014736  022777  004270  164134          CMP     #4270,@RSCS1    ;ANY ERRORS?
014744  001401                          BEQ     1$              ;NO
014746  104054                          HLT     !DA!DS!WC
014750  005777  164130          1$:     TST     @RSWC           ;DID WC GO TO 0
014754  001401                          BEQ     .+4             ;YES
014756  104010                          HLT     !WC             ;WC REG SHOULD=0
014760  006167  164210                  ROL     SAVEE           ;GET NEXT CRC WORD
014764  103404                          BCS     2$              ;DONE - BRANCH
014766  004767  010670                  JSR     PC,MDATA        ;SHIFT DATA PATTERN
014772  000167  177162                  JMP     MRCRC           ;RESTART TEST WITH NEW DATA PATTERN
014776                          2$:                             ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 55                    MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)
                                ;*****************************************************************************
014776  104400                  TST55:  SCOPE

                                ;MODULE TESTED M7753
                                ;THIS TEST IS SIMILAR TO CRC TEST 1 EXCEPT THAT THE DATA
                                ;PATTERN HAS BEEN MODIFIED TO LEAVE A SINGLE BIT SET IN THE
                                ;CRC REGISTER AFTER BOTH DATA AND CRC WORDS HAVE BEEN "READ".
                                ;THIS CAUSES A DCK ERROR.  THE READ SEQUENCE IS REPEATED 16
                                ;TIMES TO TEST THAT EACH BIT IN THE CRC REGISTER CAN CAUSE A
                                ;DCK ERROR.
015000  012767  000040  164136          MOV     #40,FLAG2       ;CLEAR TST FLAG
015006  104414          MRDCK:  CLRDK                           ;CLEAR DRIVE REGISTERS
015010  052767  000040  164152          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
015016  042767  047716  164144          BIC     #47716,ONCEE    ;CLEAR ALL OTHER FLAG BITS
015024  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
015026  104420                          MRCK                    ;CHECK MR REG
015030  022701                          22701                   ;TO EQUAL 22701
015032  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
015034  032767  000020  164102          BIT     #BIT4,FLAG2     ;FIRST TIME THROUGH
015042  001023                          BNE     3$              ;NO
015044  012767  000001  164122          MOV     #1,SAVEE        ;LOAD 1ST CRC WORD

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
                                ;CREATE BUFFER WITH 72 WORDS OF 16 BITS WHICH = THE NO. OF BITS IN 64 18 BIT WORDS
                                ;DATA BUFFER CONTAINS 7 WORDS OF ZEROS
                                ;                     A WORD OF 23
                                ;                     A WORD OF 154000
                                ;                     63 WORDS OF ZEROS
                                ;

015052  012702  026666                  MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
015056  012703  000007                  MOV     #7,R3           ;SETUP COUNTER
015062  005022          1$:     CLR     (R2)+                   ;TO CLEAR THE
015064  005303                  DEC     R3                      ;FIRST 7
015066  001375                  BNE     1$                      ;WORDS
015070  012722  000023                  MOV     #23,(R2)+       ;LOAD A 23
015074  012722  154000                  MOV     #154000,(R2)+   ;LOAD A 154000
015100  012703  000077                  MOV     #63.,R3         ;SETUP COUNTER
015104  005022          2$:     CLR     (R2)+                   ;TO CLEAR THE
015106  005303                  DEC     R3                      ;REMAINING WORDS
015110  001375                  BNE     2$                      ;FOR THAT SECTOR
                                ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0

015112  012777  027466  163766  3$:     MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
015120  012777  177700  163756          MOV     #177700,@RSWC   ;LOAD WORD COUNT REG
015126  012777  000071  163744          MOV     #71,@RSCSI      ;LOAD READ COMMAND
015134  012702  000200                  MOV     #200,R2
015140  012703  027466                  MOV     #OUTBUF,R3
015144  052767  000020  163772          BIS     #BIT4,FLAG2     ;NO SET FLAG FOR 1ST TIME THROUGH TEST
015152  005023          4$:     CLR     (R3)+
015154  005302                  DEC     R2
015156  001375                  BNE     4$
```

```
3360  015160  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
3361                                                                  ;TO CLEAR OUT COUNTERS AND REGISTERS
3362                                                                  ;THAT OTHERWISE COULD NOT BE CLEARED.
3363  015162  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
3364  015164  104456                          SPASS                   ;CLOCK MR REG SP = 1
3365                                  ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3366  015166  104430                          MRIND
3367  015170  104420                          MRCK                    ;CHECK MR REG TO EQUAL
3368  015172  022601                          22601                   ;22601
3369  015174  104000                          HLT
3370
3371                                  ;STEP THRU RESYNC PERIOD
3372
3373  015176  012767  001000  164000          MOV     #512.,REPT
3374  015204  052767  000040  163756          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT
3375  015212  104446                  MRDCK1: MCLK1                   ;CLOCK MR REG
3376  015214  104420                          MRCK                    ;CHECK FOR
3377  015216  032611                          32611                   ;CORRECT DATA
3378  015220  104000                          HLT                     ;MR=BAD GOOD=CORRECT DATA
3379  015222  104450                          MCLK0                   ;CLOCK MR REG
3380  015224  104420                          MRCK                    ;CHECK FOR
3381  015226  022601                          22601                   ;CORRECT DATA
3382  015230  104000                          HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
3383  015232  005367  163746                  DEC     REPT            ;FINISH LOOPING
3384  015236  001365                          BNE     MRDCK1          ;THROUGH RESYNC PERIOD
3385
3386                                  ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3387                                  ;SP=0 EQUALS SECTOR PULSE
3388  015240  104446                          MCLK1                   ;CLOCK MR REG
3389  015242  104420                          MRCK                    ;MR SHOULD
3390  015244  032211                          322                     ;EQUAL 32211
3391  015246  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
3392  015250  104450                          MCLK0                   ;CLOCK MR REG
3393  015252  104420                          MRCK                    ;CHECK MR
3394  015254  022201                          22201                   ;TO EQUAL 22201
3395  015256  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;PERFORM 71 MAINT CLOCK OPERATIONS--

015260  012767  000107  163716            MOV     #71.,REPT
015266  104446            MRDCK2:         MCLK1                   ;CLOCK MR REG
015270  104420                            MRCK                    ;CHECK MR REG
015272  033611                            33611                   ;TO EQUAL 33611
015274  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
015276  104450                            MCLK0                   ;CLOCK MR REG
015300  104420                            MRCK                    ;CHECK MR REG
015302  023601                            23601                   ;TO EQUAL 23601
015304  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
015306  005367  163672                    DEC     REPT            ;DONE YET
015312  001365                            BNE     MRDCK2          ;NO LOOP

                                        ;READ SYNC"1"
015314  012777  000005  163602            MOV     #5,@RSMR
015322  012777  000015  163574            MOV     #15,@RSMR
015330  104420                            MRCK
015332  133615                            133615
015334  104000                            HLT

                                        ;READ DATA
015336  005067  163664            MRDCK3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
015342  012705  026666                    MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
015346  162705  000002                    SUB     #2,R5
015352  012767  000045  163626            MOV     #45,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
015360  012767  002200  163616            MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;64 WORDS-18X64=1152
                                                                ;1 CLOCK PER 1 BIT OF DATA
015366  104444            1$:             RBIT                    ;GET 1 DATA BITS
015370  104440                            CLKR1                   ;CLOCK MR REG
015372  104000                            HLT                     ;MR NOT CORRECT

015374  104442                            CLKR0                   ;CLOCK MR REG
015376  104000                            HLT                     ;MR REG NOT CORRECT

015400  005367  163600                    DEC     REPT            ;DONE WITH COMPLETE TRANSFER
015404  001370                            BNE     1$              ;NO
```

```
 3434  015406  032767  000400  163554  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
 3435  015414  001020                          BNE     3$              ;YES
 3436  015416  052767  000400  163544          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
 3437  015424  012702  026666                  MOV     #INBUF,R2       ;MOVE CRC
 3438  015430  062702  000220                  ADD     #220,R2         ;WORD TO END OF
 3439  015434  012712  000000          4$:     MOV     #0,@R2          ;INBUF TABLE
 3440  015440  010205                  5$:     MOV     R2,R5           ;GET CRC WORD
 3441  015442  162705  000002                  SUB     #2,R5
 3442  015446  012767  000022  163530          MOV     #18.,REPT       ;SETUP TO TRANSFER 1 WD
 3443  015454  000744                          BR      1$              ;TRANSFER CRC WD
 3444  015456  104446                  3$:     MCLK1                   ;CLOCK MR REG
 3445  015460  104420                          MRCK                    ;CHECK MR REG
 3446  015462  117611                          117611                  ;TO EQUAL
 3447  015464  104000                          HLT                     ;117611
 3448  015466  104450                          MCLKO                   ;CLOCK MR REG
 3449  015470  104420                          MRCK                    ;CHECK MR
 3450  015472  003601                          3601                    ;TO EQUAL
 3451  015474  104000                          HLT                     ;3601
 3452  015476  104446                          MCLK1                   ;CLOCK MR REG
 3453  015500  104420                          MRCK                    ;CHECK MR
 3454  015502  113611                          113611                  ;TO EQUAL
 3455  015504  104000                          HLT                     ;113611
 3456  015506  104450                          MCLKO                   ;CLOCK MR REG
 3457  015510  104420                          MRCK                    ;CHECK MR
 3458  015512  003601                          3601                    ;TO EQUAL
 3459  015514  104000                          HLT                     ;3601
```

```
                                        ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
                                        ;STEP INTO END OF SECTOR DEAD BAND
                                        ;EBL IS NOW ASSERTED
                
015516  012767  000020  163460  MRDCK4: MOV     #20,REPT
                
015524  104446                  1$:     MCLK1                   ;CLOCK MR REG
015526  104420                          MRCK                    ;CHECK MR REG
015530  113611                          113611                  ;TO EQUAL
015532  104000                          HLT                     ;113611
015534  104450                          MCLKO                   ;CLOCK MR REG
015536  104420                          MRCK                    ;CHECK MR
015540  003601                          3601                    ;REG TO
015542  104000                          HLT                     ;EQUAL 3601
015544  005367  163434                  DEC     REPT            ;DONE YET?
015550  001365                          BNE     1$              ;NO
                
                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD GET STROBE BUFFER
                
015552  104446                          MCLK1                   ;CLOCK MR REG
015554  104420                          MRCK                    ;CHECK MR
015556  117611                          117611                  ;REG TO
015560  104000                          HLT                     ;EQUAL 117611
                
                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.
                
015562  104450          MRDCK5: MCLKO                           ;CLOCK MR REG
015564  022777  144270  163306          CMP     #144270,@RSCS1  ;ANY ERRORS?
015572  001401                          BEQ     1$              ;NO
015574  104054                          HLT     !DA!DS!WC
015576  005777  163302          1$:     TST     @RSWC           ;DID WC GO TO 0
015602  001401                          BEQ     .+4             ;YES
015604  104010                          HLT     !WC             ;WC REG SHOULD=0
015606  022777  100000  163300          CMP     #100000,@RSER   ;DID DCK SET?
015614  001417                          BEQ     3$              ;YES
015616  104050                          HLT     !DS!WC
015620  104402  015624                  TYPE    .+2             ;.ASCIZ <15><12>"DCK DID NOT SET "
015650  004767  004244                  JSR     PC,CRCTYP       ;GET IC THAT FAILED AND TYPE IT
015654  000241          3$:     CLC
015656  006167  163312                  ROL     SAVEE           ;GET NEXT CRC WORD
015662  103404                          BCS     2$              ;DONE - BRANCH
015664  004767  007772                  JSR     PC,MDATA        ;SHIFT DATA PATTERN
015670  000167  177112                  JMP     MRDCK           ;RESTART TEST WITH NEW DATA PATTERN
015674                  2$:             ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 56                    IGNORE FUNCTION TEST
                                ;*****************************************************************************
015674  104400                  TST56:  SCOPE

                                ;MODULE TESTED: M7759, M7770
                                ;PUT THE DISK IN MAINTENANCE MODE AND SET ERROR CONDITIONS IN THE DRIVE
                                ;ERROR REGISTER (RSER).  TRY TO START A READ TRANSFER.  THE "GO" BIT IN
                                ;RSCS1 SHOULD NOT SET.  MISSED TRANSFER ERROR (MXF) SHOULD SET IN RSCS2
                                ;WHICH IN TURN SHOULD CAUSE "TRE" AND "SC" TO SET IN RSCS1.

015676  104414          MRIFT:  CLRDK                   ;CLEAR ALL REGISTERS
015700  104416                  MRDMD                   ;PUT DRIVE INTO MAINT MODE
015702  104420                  MRCK                    ;CHECK MR REG
015704  022701                  22701                   ;TO EQUAL 22701
015706  104424                  MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
015710  012777  177777  163176  MOV     #-1,@RSER       ;SET ERRORS
015716  016777  163242  163172  MOV     UNITSV,@RSAS    ;CLEAR ATA BIT IN RSAS
                                                        ;AND ERROR BITS IN RSCS1
015724  012777  027466  163154  MOV     #OUTBUF,@RSBA   ;LOAD RSBA
015732  012777  177777  163144  MOV     #-1,@RSWC       ;LOAD RSWC
015740  012777  000071  163132  MOV     #71,@RSCS1      ;LOAD READ FUNCTION
015746  032777  000001  163124  BIT     #BIT0,@RSCS1    ;IS "GO" BIT ZERO?
015754  001401                  BEQ     1$              ;YES
015756  104140                  HLT     !DS!AS          ;"GO" BIT IN RSCS1 SHOULD NOT
                                                        ;LOAD IF ERRORS ARE PRESENT IN THE DRIVE
015760  012767  177777  163230  1$:     MOV     #177777,WORK    ;SETUP TIMEOUT FOR MXF ERROR
015766  005367  163224          5$:     DEC     WORK
015772  000240                  NOP
015774  000240                  NOP
015776  001373                  BNE     5$
016000  017700  163076          MOV     @RSCS2,BAD      ;CHECK RSCS2 FOR MXF
016004  012701  001100          MOV     #1100,GOOD      ;GET CORRECT ANS
016010  056701  163146          BIS     UNNUM,GOOD      ;FOR RSCS2
016014  020001                  CMP     BAD,GOOD        ;IS RSCS2 CORRECT
016016  001401                  BEQ     2$              ;YES
016020  104000                  HLT                     ;BAD=RSCS2 GOOD=CORRECT ANS
                                                        ;MXF SHOULD BE SET IN RSCS2
                                                        ;FOR A READ WAS ISSUED
                                                        ;WITH ERROR BITS SET IN RSER.
016022  022777  144270  163050  2$:     CMP     #144270,@RSCS1  ;IS RSCS1 CORRECT?
016030  001401                  BEQ     3$              ;YES
016032  104042                  HLT     !DS!ER          ;SC AND TRE SHOULD BE SET FOR
                                                        ;MXF SHOULD BE SET IN RSCS2
```

```
3550  016034  022777  177777  163042  3$:   CMP   #-1,@RSWC      ;DID RSWC CHANGE?
3551  016042  001401                        BEQ   .+4             ;NO
3552  016044  104010                        HLT   !WC             ;WC SHOULD NOT HAVE CHANGED
3553                                                              ;FOR READ SHOULD HAVE NEVER
3554                                                              ;BEEN EXECUTED
3555  016046  022777  027466  163032        CMP   #OUTBUF,@RSBA   ;DID RSBA MOVE
3556  016054  001405                        BEQ   4$              ;NO
3557  016056  017700  163024               MOV   @RSBA,BAD       ;GET RSBA
3558  016062  012701  027466               MOV   #OUTBUF,GOOD    ;GET CORRECT ANS
3559  016066  104000                        HLT                   ;RSBA=BAD GOOD=CORRECT ANS
3560                                                              ;RSBA MOVED WHEN THE READ
3561                                                              ;COMD WAS LOADED WITH ERROR
3562                                                              ;BITS SET IN RSER, READ COMD SHOULD
3563                                                              ;NEVER HAVE BEEN EXECUTED AND RSBA
3564                                                              ;SHOULD NOT HAVE MOVED.
3565  016070  104414                4$:   CLRDK                   ;CLEAR ALL REGISTERS
3566  016072  017700  163004               MOV   @RSCS2,BAD      ;GET RSCS2
3567  016076  012701  000100               MOV   #100,GOOD       ;SETUP CORRECT
3568  016102  056701  163054               BIS   UNNUM,GOOD      ;ANS
3569  016106  020100                        CMP   GOOD,BAD        ;IS CS2 CORRECT?
3570  016110  001401                        BEQ   .+4             ;YES
3571  016112  104000                        HLT                   ;MXF SHOULD HAVE CLEARED IN
3572                                                              ;RSCS2 WITH THE "CLR" BIT BEING
3573                                                              ;SET IN RSCS2.
```

```
                                ;*****************************************************************************
                                ;TEST 57                    INVALID ADDRESS ERROR (IAE) TEST
                                ;*****************************************************************************
016114  104400                  TST57:  SCOPE

                                ;MODULE TESTED M7754, M7770
                                ;FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK
                                ;ADDRESS REGISTER (RSDA).  THIS SHOULD CAUSE "IAE" TO SET IN
                                ;THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO
                                ;RSCS1 WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE
                                ;DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE
                                ;CONTROL REGISTER (RSCS1).
016116  042767  000040  163044          BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
016124  012702  004000                  MOV     #4000,R2        ;LOAD R2 WITH INVALID ADDR
016130  012767  016136  162652          MOV     #4$,LAD         ;LOOP HERE ON ERROR
016136  104416          4$:     MRDMD                           ;PUT DRIVE IN MAINT MODE
016140  104420                  MRCK                            ;CHECK MAINT REG
016142  022701                  22701
016144  104424                  MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
016146  032767  000004  163014          BIT     #BIT2,ONCEE     ;LOOPING ON ERRORS>
016154  001002                  BNE     1$                      ;YES
016156  006102                  ROL     R2                      ;GET INVALID ADDRESS
016160  103454                  BCS     IADONE                  ;DONE FLOATING A ONE YET?
016162  010277  162722          1$:     MOV     R2,@RSDA        ;LOAD RSDA WITH INVALID ADDRESS
016166  012777  000071  162704          MOV     #71,@RSCS1      ;DO A READ TO INVALID ADDR
016174  022777  002000  162712          CMP     #2000,@RSER     ;IS RSER CORRECT?
016202  001404                  BEQ     2$                      ;YES
016204  052767  000004  162756          BIS     #BIT2,ONCEE     ;SET ERROR BIT
016212  104044                  HLT     !DS!DA                  ;RSER SHOULD=2000 FOR
                                                                ;A READ COMMAND WAS GIVEN
                                                                ;TO AN ILLEGAL ADDRESS
016214  042767  000004  162746  2$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
016222  022777  150600  162662          CMP     #150600,@RSDS   ;DID IAE SET?
016230  001404                  BEQ     3$                      ;YES
016232  052767  000004  162730          BIS     #BIT2,ONCEE     ;SET ERROR BIT
016240  104044                  HLT     !DS!DA                  ;RSDS SHOULD=150600 FOR
                                                                ;IAE SHOULD BE SET IN RSER
016242  042767  000004  162720  3$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
016250  032777  100000  162622          BIT     #BIT15,@RSCS1   ;DID SC SET?
016256  001004                  BNE     5$                      ;YES
016260  052767  000004  162702          BIS     #BIT2,ONCEE     ;SET ERROR BIT
016266  104044                  HLT     !DA!DS                  ;SC SHOULD BE SET IN RSCS1
                                                                ;FOR IAE SHOULD BE SET IN RSER
016270  042767  000004  162672  5$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR BIT
016276  104414                  CLRDK                           ;CLEAR ALL ERRORS
016300  005777  162610                  TST     @RSER           ;DID IAE CLEAR?
016304  001401                  BEQ     .+4                     ;YES
016306  104040                  HLT     !DS                     ;IAE DID NOT CLEAR
016310  000712                  BR      4$                      ;CONTINUE
016312                          IADONE:                         ;DONE
```

```
                        ;*****************************************************************************
                        ;TEST 60                    OPERATION INCOMPLETE ERROR TEST
                        ;*****************************************************************************
016312  104400          TST60:  SCOPE

                        ;MODULE TESTED M7770
                        ;PUT THE DISK IN MAINTENANCE MODE AND START A READ COMMAND
                        ;THEN ISSUE THREE DISK "INDEX" PULSES TO SIMULATE A COMPLETE
                        ;ROTATION OF THE DISK SURFACE.  THE THIRD INDEX PULSE SHOULD
                        ;CAUSE OPERATION  IN COMPLETE "OPI" TO APPEAR IN THE DRIVE ERROR
                        ;REGISTER (RSER) AND "ATA" AND "ERR" IN THE DRIVE STATUS REGISTER (RSDS)

016314  104414          MROPI:  CLRDK                           ;CLEAR ALL DRIVE REGISTERS
016316  013777  027466  162562          MOV     @#OUTBUF,@RSBA  ;SETUP RSBA
016324  012777  177777  162552          MOV     #-1,@RSWC       ;SETUP RSWC

016332  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
016334  104420                          MRCK                    ;CHECK MAINT REG
016336  022701                          22701                   ;TO EQUAL 22701
016340  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)

016342  012777  000071  162530          MOV     #71,@RSCS1      ;LOAD A READ COMMAND

016350  104430                          MRIND                   ;ISSUE THREE INDEX
016352  104430                          MRIND                   ;PULSES TO
016354  104430                          MRIND                   ;CAUSE OPI

                        ;NOW CHECK FOR CORRECT ERRORS IN RSER AND RSDS
016356  017700  162532                  MOV     @RSER,BAD       ;GET RSER
016362  012701  020000                  MOV     #20000,GOOD     ;GET CORRECT ANS
016366  020100                          CMP     GOOD,BAD        ;DID OPI SET IN RSER?
016370  001434                          BEQ     1$              ;YES
016372  104402  016376                  TYPE    ,.+2            ;.ASCIZ <15><12>"OPI IN RSER SHOULD SET-3 INDEX PULSES W
016460  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS

016462  022777  150600  162422  1$:     CMP     #150600,@RSDS   ;DID CORRECT ERRORS SET?
016470  001401                          BEQ     2$              ;YES
016472  104040                          HLT     !DS             ;RSDS SHOULD=150600 BECAUSE
                                                                ;OF OPI ERROR IN RSER
016474  022777  144270  162376  2$:     CMP     #144270,@RSCS1  ;DID SC AND TRE SET IN RSCS1?
016502  001401                          BEQ     MROPIA          ;YES
016504  104050                          HLT     !DS!WC          ;SC AND TRE SHOULD SET IN RSCS1
                                                                ;BECAUSE OF ERROR IN RSER
016506  104414          MROPIA: CLRDK                           ;CLEAR ALL ERRORS
016510  005777  162400                  TST     @RSER           ;DID OPI CLEAR IN RSER ?
016514  001437                          BEQ     1$              ;YES
016516  104402  016522                  TYPE    .+2             ;.ASCIZ <15><12>"OPI IN RSER DID NOT CLEAR BY SETTING CL
016612  104040                          HLT     !DS             ;RSER SHOULD=0
016614  022777  010600  162270  1$:     CMP     #10600,@RSDS    ;DID ERROR BITS CLEAR IN RSDS
                                                                ;BY SETTING CLR BIT IN RSCS2
016622  001401                          BEQ     .+4             ;YES
016624  104040                          HLT     !DS             ;RSDS SHOULD=10600
```

```
                                ;*****************************************************************************
                                ;TEST 61                    PARITY ERROR TEST
                                ;*****************************************************************************
016626  104400                  TST61:  SCOPE

                                ;MODULES TESTED: M7754, M7770
                                ;SET "PAT" BIT IN RSCS2.  WRITE A DRIVE REGISTER.  "PAR" SHOULD SET IN
                                ;THE DRIVE ERROR REGISTER (RSER) WHICH SHOULD CAUSE "ATA" TO SET IN RSAS
                                ;AND 'SC' TO SET IN RSCS1.

016630  104414                  MRPAR:  CLRDK                   ;CLEAR ALL REGISTERS
016632  042767  000040  162330          BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
016640  104416                          MRDMD                   ;PUT DRIVE IN MAINT MODE
016642  104420                          MRCK                    ;CHECK MAINT TO
016644  022701                          22701                   ;EQUAL 22701
016646  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
016650  052777  000020  162224          BIS     #BIT4,@RSCS2    ;SET THE "PAT" BIT.
016656  012777  000077  162224          MOV     #77,@RSDA       ;BY WRITING INTO THIS REGISTER,
                                                                ;PAR SHOULD SET IN RSER
016664  022777  000010  162222          CMP     #10,@RSER       ;DID PAR SET?
016672  001401                          BEQ     .+4             ;YES
016674  104040                          HLT     !DS             ;"PAR" IN RSER SHOULD BE SET FOR
                                                                ;THE "PAT" BIT WAS SET IN RSCS2
                                                                ;WHEN PROGRAM TRIED TO WRITE INTO RSDA
016676  022777  104200  162174          CMP     #104200,@RSCS1  ;DID PAR CAUSE SC TO SET?
016704  001401                          BEQ     .+4             ;YES
016706  104044                          HLT     !DS!DA          ;SC SHOULD BE SET IN RSCS1 FOR
                                                                ;PAR SHOULD BE SET IN RSER
016710  022777  000077  162172          CMP     #77,@RSDA       ;DID RSDA GET LOADED?
016716  001401                          BEQ     .+4             ;YES
016720  104004                          HLT     !DA             ;RSDA SHOULD=77 FOR PAT
                                                                ;BIT WAS SET WHEN PROGRAM
                                                                ;TRIED TO WRITE INTO RSDA
016722  104414                          CLRDK                   ;CLEAR ALL ERRORS
016724  022777  004200  162146          CMP     #4200,@RSCS1    ;DID ERRORS CLEAR?
016732  001401                          BEQ     .+4             ;YES
016734  104044                          HLT     !DS!DA          ;SC DID NOT CLEAR BY USING
                                                                ;THE "CLR" BIT IN RSCS2
016736  005777  162152                  TST     @RSER           ;DID PAR CLEAR?
016742  001401                          BEQ     .+4             ;YES
016744  104044                          HLT     !DS!DA          ;PAR DID NOT CLEAR BY USING
                                                                ;THE CLR BIT IN RSCS2
```

```
3718                                    ;*************************************************************************
3719                                    ;TEST 62                MAINTENANCE MODE INTERRUPT TEST
3720                                    ;*************************************************************************
3721  016746  104400                    TST62:  SCOPE
3722
3723                                    ;MODULE TESTED M7771
3724                                    ;IN THIS TEST THE INTERRUPT ENABLE BIT IS SET (I.E.).
3725                                    ;A TWO SECTOR WRITE COMMAND IS GIVEN.  AN "RMR"
3726                                    ;ERROR IS CREATED WHILE THE FIRST SECTOR IS BEING WRITTEN
3727                                    ;THIS SHOULD CAUSE THE DRIVE TO INTERRUPT AFTER THE FIRST
3728                                    ;SECTOR IS WRITTEN. AND CAUSE THE TRANSFER TO TERMINATE
3729
3730  016750  012767  001602  162166          MOV     #1602,FLAG2
3731  016756  104414            MREX:   CLRDK                   ;CLEAR DRIVE REGISTERS
3732  016760  012737  000200  177776          MOV     #200,@#PS       ;SETUP FOR INTERRUPT
3733  016766  012706  000500                  MOV     #500,SP
3734  016772  012767  000040  162170          MOV     #40,ONCEE       ;SET TYPE CLOCK CNT WITH ERROR MESSAGE FLAG
3735  017000  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
3736  017002  104420                          MRCK                    ;CHECK MR REG
3737  017004  022701                          22701                   ;TO EQUAL 22701
3738  017006  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
3739                                                                  ;BY SENDING 2 CLOCK PULSES
3740
3741                                    ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (2 SECTORS)
3742                                    ;DATA BUFFER WORDS ARE  :A WORD OF ALL 0'S - ALL 1'S
3743                                    ;                       FLOATING 1'S PATTERN (16 WORDS)
3744                                    ;                       A PATTERN OF 146314 (110 WORDS)
3745
3746  017010  012702  026666                  MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
3747  017014  005022                          CLR     (R2)+           ;CLEAR 1ST LOCATION
3748  017016  012722  177777                  MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
3749  017022  005003                          CLR     R3              ;CLEAR WORK LOC TO GENERATE
3750  017024  000261                          SEC                     ;A PATTERN OF FLOATING ONES
3751  017026  006103            1$:     ROL     R3                      ;GET PATTERN
3752  017030  103402                    BCS     2$                      ;DONE GET OUT
3753  017032  010322                    MOV     R3,(R2)+                ;FILL BUFFER
3754  017034  000774                    BR      1$                      ;CONT
3755  017036  012703  000156    2$:     MOV     #110.,R3                ;FILL REMAINING PORTION OF
3756  017042  012704  146314            MOV     #146314,R4              ;BUFFER WITH A PATTERN OF 146314
3757  017046  010422            3$:     MOV     R4,(R2)+                ;LOAD BUFFER
3758  017050  005303                    DEC     R3                      ;DONE YET?
3759  017052  001375                    BNE     3$                      ;NO
3760
3761                                    ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (2 SECTORS)
3762  017054  012777  017674  162046          MOV     #INTMR,@RSVEC   ;SETUP INTERRUPT VECTOR
3763  017062  012777  000340  162042          MOV     #340,@RSVCPS
3764  017070  012777  026666  162010          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
3765  017076  012777  177600  162000          MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
3766  017104  012777  000161  161766          MOV     #161,@RSCS1     ;LOAD WRITE COMMAND I/E
3767  017112  104454                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
3768                                                                  ;TO CLEAR OUT COUNTERS AND REGISTERS
3769                                                                  ;THAT OTHERWISE COULD NOT BE CLEARED.
3770  017114  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
3771  017116  104456                          SPASS                   ;CLOCK MR REG SP = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
017120  104430                                MRIND
017122  104420                                MRCK                    ;CHECK MR REG TO EQUAL
017124  020501                                20501                   ;20501
017126  104000                                HLT

                                        ;STEP THRU RESYNC PERIOD

017130  012767  001000  162046                MOV     #512.,REPT
017136  052767  000040  162024                BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
017144  104446                        MREX1:  MCLK1                   ;CLOCK MR REG
017146  104420                                MRCK                    ;CHECK FOR
017150  030511                                30511                   ;CORRECT DATA
017152  104000                                HLT                     ;MR = BAD GOOD = CORRECT DATA
017154  104450                                MCLK0                   ;CLOCK MR REG
017156  104420                                MRCK                    ;CHECK FOR
017160  020501                                20501                   ;CORRECT DATA
017162  104000                                HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
017164  005367  162014                        DEC     REPT            ;FINISH LOOPING
017170  001365                                BNE     MREX1           ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
017172  104446                                MCLK1                   ;CLOCK MR REG
017174  104420                                MRCK                    ;MR SHOULD
017176  030111                                30111                   ;EQUAL 30111
017200  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
017202  104450                                MCLK0                   ;CLOCK MR REG
017204  104420                                MRCK                    ;CHECK MR
017206  020101                                20101                   ;TO EQUAL 20101
017210  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 63 MAINT CLOCK OPERATIONS--WRITING PREAMBLE

017212  012767  000077  161764                MOV     #63.,REPT
017220  104446                        MREX2:  MCLK1                   ;CLOCK MR REG
017222  104420                                MRCK                    ;CHECK MR REG
017224  031511                                31511                   ;TO EQUAL 31511
017226  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
017230  104450                                MCLK0                   ;CLOCK MR REG
017232  104420                                MRCK                    ;CHECK MR REG
017234  021501                                21501                   ;TO EQUAL 21501
017236  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
017240  005367  161740                        DEC     REPT            ;DONE YET
017244  001365                                BNE     MREX2           ;NO LOOP
```

```
                                        ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
017246  104446                                MCLK1                   ;CLOCK MR REG
017250  104420                                MRCK                    ;CHECK MR REG
017252  131511                                131511                  ;TO EQUAL 131511
017254  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017256  104450                                MCLK0                   ;CLOCK MR REG
017260  104420                                MRCK                    ;MR REG SHOULD
017262  025501                                25501                   ;EQUAL 25501
017264  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017266  104446                                MCLK1
017270  104420                                MRCK
017272  135511                                135511
017274  104000                                HLT
                                        ;PERFORM NEXT STEP 8 TIMES TO FINISH WRITING PREAMBLE
017276  012767  000010  161700                MOV     #10,REPT
017304  104450                        MREX3:  MCLK0                   ;CLOCK MR REG
017306  104420                                MRCK                    ;CHECK MR REG
017310  025501                                25501                   ;TO EQUAL 25501
017312  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
017314  104446                                MCLK1                   ;CLOCK MR REG
017316  104420                                MRCK                    ;CHECK MR REG
017320  135511                                135511                  ;TO EQUAL 135511
017322  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017324  005367  161654                        DEC     REPT            ;DONE YES?
017330  001365                                BNE     MREX3           ;NO LOOP BACK

                                        ;MOVE DATA WORD INTO RS03 SHIFT REGISTER

017332  104450                                MCLK0                   ;CLOCK MR REG
017334  104420                                MRCK                    ;CHECK MR REG
017336  021501                                21501                   ;TO EQUAL 21501
017340  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
017342  104446                                MCLK1                   ;CLOCK MR REG
017344  104420                                MRCK                    ;MR REG SHOULD
017346  123511                                123511                  ;EQUAL 123511
017350  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;ENCODE SYNC 1

017352  104450                                MCLK0                   ;CLOCK MR REG
017354  104420                                MRCK                    ;MR REG SHOULD NOW
017356  033501                                33501                   ;EQUAL 33501
017360  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
017362  012705  026666                        MOV     #INBUF,R5       ;GET STARTING ADDR FOR DATA BUFFER
017366  011504                                MOV     (R5),R4         ;GET DATA
```

```
3863  017370  012767  002156  161620          MOV     #1134.,WORK     ;DOING A 1 SECTOR TRANSFER 127 WORDS
3864                                                                    ;18 BITS PER WORD-CLOCK LOOPS
3865                                                                    ;TAKE CARE OF 2 BITS AT A TIME
3866                                                                    ;64 TIMES 18 EQUALS 1134 LOOPS
3867                                                                    ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
3868  017376  052767  000100  161564          BIS     #BIT6,ONCEE     ;SET 1ST TRANSFER WORD FLAG
3869  017404  104432                  1$:     XBIT                    ;GET 1 BIT OF DATA
3870  017406  104434                          CLKD1                   ;SET MCLK
3871                                                                    ;AND CALCULATE MR REG
3872                                                                    ;FOR CORRECT DATA (MWDB)
3873  017410  104000                          HLT                     ;MR REG NOT CORRECT
3874  017412  104436                          CLKD0                   ;CLEAR MCLK TO
3875                                                                    ;COMPLETE TRANSFER OF 1 BIT
3876                                                                    ;CALCULATE CORRECT ANS FOR
3877                                                                    ;MR REG (MWDB)
3878  017414  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
3879  017416  032767  000200  161544          BIT     #BIT7,ONCEE     ;ON LAST WORD YET?
3880  017424  001015                          BNE     2$              ;YES
3881  017426  032767  000400  161534          BIT     #BIT8,ONCEE     ;ON CRC WORD YET?
3882  017434  001043                          BNE     3$              ;YES
3883  017436  005367  161554                  DEC     WORK            ;DONE WITH 63 WORDS?
3884  017442  001360                          BNE     1$              ;NO
3885
3886  017444  052767  000200  161516          BIS     #BIT7,ONCEE     ;SET LAST WORD FLAG
3887  017452  012767  000023  161536          MOV     #19.,WORK       ;SET UP TO TRANSFER LAST WORD
3888  017460  005367  161532          2$:     DEC     WORK            ;DONE YET
3889  017464  001347                          BNE     1$
3890
3891  017466  052767  000400  161474          BIS     #BIT8,ONCEE     ;SET TRANSFERRING CRC WORD
3892  017474  042767  000200  161466          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
3893
3894                          ;GENERATE RMR ERROR BY ATTEMPTING TO WRITE RSER
3895                          ;EXC SHOULD THEN BE ASSERTED
3896
3897  017502  012777  177777  161404          MOV     #-1,@RSER
3898  017510  004767  004026                  JSR     PC,GENCRC       ;GENERATE CRC WORD
3899                                                                    ;AND LEAVE IN "WORK"
3900  017514  012702  026666                  MOV     #INBUF,R2       ;GO TO END
3901  017520  062702  000200                  ADD     #200,R2         ;OF DATA BUFFER
3902  017524  016712  161466                  MOV     WORK,@R2        ;LOAD CRC WORD
3903  017530  010205                          MOV     R2,R5           ;RESET POINTER FOR
3904  017532  162705  000002                  SUB     #2,R5           ;R5 FOR CRC WD
3905  017536  012767  000023  161452          MOV     #19.,WORK       ;SETUP TO XFER CRC
3906  017544  005367  161446          3$:     DEC     WORK            ;DONE YET?
3907  017550  001315                          BNE     1$              ;NO
```

```
                                        ;EBL SHOULD NOW ASSERT AND CRC BE WRITTEN

017552  104446                                  MCLK1                   ;CLOCK MR REG TO STEP THROUGH DEAD BAND AREA
017554  104420                                  MRCK                    ;CHECK MR REG
017556  113511                                  113511                  ;TO EQUAL 113511
017560  104000                                  HLT                     ;MR REG=BAD GOOD=CORRECT ANS

                                        ;LOOP 17 TIMES

017562  012767  000017  161414                  MOV     #17,REPT
017570  104450                          4$:     MCLK0                   ;CLOCK MR REG
017572  104420                                  MRCK                    ;CHECK MR REG
017574  003501                                  3501                    ;TO EQUAL 3501
017576  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017600  104446                                  MCLK1                   ;CLOCK MR REG
017602  104420                                  MRCK                    ;CHECK MR REG
017604  113511                                  113511                  ;TO EQUAL 11[illegible]11
017606  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017610  005367  161370                          DEC     REPT            ;DONE LOOPING YET?
017614  001365                                  BNE     4$              ;NO

                                        ;FINISH UP

017616  104450                                  MCLK0                   ;CLOCK MR REG
017620  104420                                  MRCK                    ;CHECK MR REG
017622  003501                                  3501                    ;TO EQUAL 3501
017624  104000                                  HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017626  104446                                  MCLK1                   ;CLOCK MR REG
017630  104420                                  MRCK                    ;CHECK MR REG
017632  111511                                  111511                  ;TO EQUAL 111511
017634  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;TRANSFER SHOULD NOW BE COMPLETE

017636  104450                                  MCLK0                   ;CLOCK MR REG
017640  104420                                  MRCK                    ;CHECK MRE REG
017642  001501                                  1501                    ;TO EQUAL 1501
017644  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017646  104446                                  MCLK1                   ;CLOCK MR REG
017650  104420                                  MRCK                    ;CHECK MR
017652  012711                                  12711                   ;TO EQUAL
017654  104000                                  HLT                     ;12711
017656  104450                                  MCLK0                   ;CLOCK MR
017660  104420                                  MRCK                    ;CHECK MR
017662  002201                                  2201                    ;TO EQUAL
017664  104000                                  HLT                     ;2201
017666  000240                                  NOP                     ;STALL FOR TIME
017670  104050                                  HLT     !WC!DS          ;SHOULD NEVER GET HERE
017672  000424                                  BR      INTMR1          ;BECAUSE DRIVE SHOULD HAVE INTERRUPTED,
                                                                        ;CAUSING JUMP TO INTMR.
                                                                        ;CHECK FOR ASSERTION OF FTS ATTN L.
```

```
                                        ;NOW TEST CONTROLLER
 3959
 3960
 3961  017674  022777  144260  161176  INTMR:  CMP     #144260,@RSCS1  ;IS CS1 CORRECT?
 3962  017702  001401                          BEQ     .+4             ;YES
 3963  017704  104014                          HLT     !DA!WC          ;YES
 3964  017706  022777  000001  161174  5$:     CMP     #1,@RSDA        ;IS RSDA CORRECT?
 3965  017714  001401                          BEQ     .+4             ;YES
 3966  017716  104004                          HLT     !DA             ;DA SHOULD = 1
 3967  017720  022777  000004  161166          CMP     #4,@RSER        ;DID RMR SET IN RSER
 3968  017726  001401                          BEQ     .+4             ;YES
 3969  017730  104050                          HLT     !DS!WC          ;RSER SHOULD = 4
 3970  017732  022777  000001  161150          CMP     #1,@RSDA        ;DOES RSDA=1
 3971  017740  001401                          BEQ     .+4             ;YES
 3972  017742  104004                          HLT     !DA             ;RSDA SHOULD=1
 3973  017744  000240                  INTMR1: NOP                     ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 63                    DISK ADDRESS OVERFLOW TEST
                                ;*****************************************************************************
017746  104400                  TST63:  SCOPE

                                ;MODULES TESTED: M7754, M7771, M7770
                                ;SET UP TO TRANSFER 2 SECTORS TO THE DISK, STARTING AT TRACK 77 SECTOR 77
                                ;TO CAUSE A DISK ADDRESS OVERFLOW CONDITION. ALSO CHECK LAST BLOCK TRANSFER
                                ;(LBT) BIT TO SET IN THE RSDS REGISTER.

017750  104414                  MRAOE:  CLRDK                   ;CLEAR ALL REGISTERS
017752  012706  000500                  MOV     #500,SP         ;SETUP STACK POINTER
017756  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
017760  104420                          MRCK                    ;CHECK MAINT REG
017762  022701                          22701                   ;TO EQUAL 22701
017764  104424                          MRINT                   ;INITIALIZE MAINT REG BY SENDING
                                                                ;2 CLOCK PULSES (CLEAR MRSP)
017766  012777  007777  161114          MOV     #7777,@RSDA     ;SETUP DISK ADDRESS
017774  012777  177400  161102          MOV     #-400,@RSWC     ;SETUP FOR A 2 SECTOR TRANSFER
020002  012777  027466  161076          MOV     #OUTBUF,@RSBA   ;GET OUTPUT BUFFER

                                ;SETUP BUFFER WITH ALL ONES
020010  012705  027466                  MOV     #OUTBUF,R5      ;GET STARTING ADDRESS OF OUTBUF
020014  012767  000400  161162          MOV     #400,REPT       ;LOAD 2 SECTORS
020022  012725  177777          1$:     MOV     #-1,(R5)+       ;WITH WORDS
020026  005367  161152                  DEC     REPT            ;OF ALL ONES
020032  001373                          BNE     1$

020034  012777  000061  161036          MOV     #61,@RSCS1      ;LOAD WRITE COMMAND
020042  104430                          MRIND                   ;SET INDEX PULSE

                                ;SUPPLY CLOCKS TO STEP THROUGH A TRACK

020044  012767  000002  161132          MOV     #2,REPT
020052  012704  124000          5$:     MOV     #43008.,R4      ;SETUP FOR FAST CLOCK PULSES 172032 CLOCKS
020056  012702  000011                  MOV     #11,R2
020062  012703  000001                  MOV     #1,R3
020066  010277  161032          2$:     MOV     R2,@RSMR
020072  010377  161026                  MOV     R3,@RSMR
020076  005304                          DEC     R4
020100  001372                          BNE     2$
020102  005367  161076                  DEC     REPT
020106  001361                          BNE     5$

020110  104422                          MRCLK                   ;CLOCK A 11 AND A 1 INTO RSMR
020112  104426                          DSCK                    ;CHECK MR
020114  012400                          12400                   ;TO EQUAL 12400
020116  104000                          HLT                     ;LBT SHOULD BE SET IN RSDS
```

```
                                        ;ASSERT MAINTENANCE INDEX PULSE TO RESET DRIVE
                                        ;FOR THE SECOND REVOLUTION

020120  104430                          MRIND                   ;ASSERT MAINT INDEX PULSE
020122  005067  161050                  CLR     MCCNT           ;CLEAR THE CLOCK COUNTER
020126  104420                          MRCK                    ;CHECK MR REG
020130  002501                          2501                    ;TO EQUAL 2501. SHOULD STILL BE WRITING
020132  104000                          HLT

                                        ;SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS03 RESYNC PERIOD
020134  012767  001000  161042          MOV     #512.,REPT      ;CLOCK COUNT TO STEP THRU RESYNC
020142  104446          4$:             MCLK1                   ;2ND REVOLUTION
020144  104420                          MRCK                    ;CHECK MR
020146  012511                          12511                   ;TO EQUAL 12511
020150  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
020152  104450                          MCLKD                   ;CLOCK MR REG
020154  104420                          MRCK                    ;CHECK MR
020156  002501                          2501                    ;REG TO
020160  104000                          HLT                     ;EQUAL 2501
020162  005367  161016                  DEC     REPT
020166  001365                          BNE     4$              ;LOOP TILL DONE

                                        ;SUPPLY 2 CLOCKS TO CAUSE THE SECTOR PULSE TO APPEAR IN
                                        ;THE MR REGISTER AND THE "AOE" ERROR TO APPEAR IN
                                        ;THE RSER REGISTER

020170  104422          AOECK:  MRCLK
020172  104422                  MRCLK                           ;CAUSE SECTOR PULSE AND AOE ERROR
020174  104420                  MRCK                            ;CHECK FOR SECTOR PULSE
020176  022701                  22701                           ;IN RSMR
020200  104000                  HLT                             ;MR=BAD GOOD=CORRECT ANS
020202  022777  001000  160704          CMP     #1000,@RSER     ;DID AOE SET IN RSER?
020210  001401                  BEQ     1$                      ;AOE SHOULD BE SET IN RSER
020212  104040                  HLT     !DS                     ;RSER SHOULD EQUAL 1000
020214  022777  152600  160670  1$:     CMP     #152600,@RSDS   ;IS RSDS CORRECT
020222  001401                  BEQ     2$                      ;YES
020224  104040                  HLT     !DS                     ;ERR & ATA SHOULD BE SET IN RSDS
                                                                ;BECAUSE OF AOE ERROR IN RSER
020226  104414          2$:     CLRDK                           ;CLEAR ERROR
020230  005777  160660          TST     @RSER                   ;DID ERROR CLEAR?
020234  001401                  BEQ     3$                      ;YES
020236  104040                  HLT     !DS                     ;AOE DID NOT CLEAR BY SETTING CLR IN RSCS2
020240  022777  010600  160644  3$:     CMP     #10600,@RSDS    ;DID ERRORS CLEAR
020246  001401                  BEQ     .+4                     ;YES
020250  104040                  HLT     !DS                     ;ERR AND ATA & LBT SHOULD ALL BE CLEARED
                                                                ;FOR CLR WAS SET IN RSCS2
```

```
                                ;MAINTENANCE MODE VERIFY TEST
                                ;-----DANGER---THIS TEST DESTROYS DATA ON DISKS--DANGER
                                ;THIS TEST WILL ONLY RUN IF SWITCH 11 IS SET IN THE "SWITCH
                                ;REGISTER" FOR IT WILL ACTUALLY WRITE DATA ONTO THE DISK.  IT
                                ;WILL WRITE ONE TRACK OF ALL ONES. THE PROGRAM THEN GOES BACK
                                ;TO THE MAINT WRITE TEST AND WRITES ONE SECTOR OF DATA (ZERO'S, ONES, FLOATING
                                ;ONES AND FILLS THE REMAINDER OF SECTOR WITH A PATTERN OF 146314)
                                ;THE DRIVE IS THEN TAKEN OUT OF
                                ;"MAINTENANCE MODE" AND THE TRACK IS THEN READ. THE TRACK
                                ;SHOULD CONTAIN ALL ONES.

                                ;*************************************************************************
                                ;TEST 64                  MAINTENANCE MODE VERIFY TEST
                                ;*************************************************************************
020252  104400                  TST64:  SCOPE

                                ;MODULE TESTED G182

020254  032767  004000  157306  MRVR:   BIT     #BIT11,SWR      ;DO THIS TEST?
020262  001002                          BNE     3$              ;YES
020264  000137  020774                   JMP     @#INFTST        ;NO
020270  012767  001600  160646  3$:     MOV     #1600,FLAG2     ;SET VERIFY TEST FLAG
020276  104414                          CLRDK                   ;CLEAR ALL DRIVES
020300  012767  177777  160724          MOV     #177777,WORK5   ;STALL TO RESYNC
020306  005367  160720          4$:     DEC     WORK5           ;DRIVE
020312  001375                          BNE     4$              ;TIMING LOGIC

                                ;STEP THRU RESYNC PERIOD

020314  012777  170000  160562          MOV     #-10000,@RSWC   ;WRITE ONE TRACK - 4K WORDS
020322  012767  177777  006336          MOV     #177777,INBUF   ;WRITE A PATTERN 12525
020330  052777  000010  160544          BIS     #BIT3,@RSCS2    ;SET BAI BIT
020336  012777  026666  160542          MOV     #INBUF,@RSBA    ;SET DATA WD
020344  012767  177777  160632          MOV     #177777,REPT    ;SETUP WAIT LOOP
020352  012777  000061  160520          MOV     #61,@RSCS1      ;GO WRITE
020360  105777  160514          1$:     TSTB    @RSCS1          ;DONE YET?
020364  100404                          BMI     2$              ;YES
020366  005367  160612                  DEC     REPT            ;DECREMENT COUNTER WAITING
020372  001372                          BNE     1$              ;FOR READY
020374  104000                          HLT                     ;READY NEVER CAME UP
020376  005777  160476          2$:     TST     @RSCS1          ;ANY ERRORS?
020402  100002                          BPL     MRVR1           ;NO
020404  104050                          HLT     !DS!WC          ;STOP HERE TILL THIS PROBLEM IS FIXED TRY DZRSB DIAG
020406  000433                          BR      TBDIA           ;TYPE MESSAGE
```

```
 4112  020410  104414                    MRVR1:  CLRDK                   ;CLEAR ALL REGISTERS
 4113  020412  012777  170000  160464            MOV     #-10000,@RSWC   ;SETUP WC
 4114  020420  052777  000010  160454            BIS     #BIT3,@RSCS2    ;SET BAI
 4115  020426  012777  026666  160452            MOV     #INBUF,@RSBA    ;SETUP RSBA
 4116  020434  012767  177777  160542            MOV     #177777,REPT    ;SETUP WAIT LOOP
 4117  020442  012777  000051  160430            MOV     #51,@RSCS1      ;DO A WRITE CHECK TO VERIFY DISK
 4118  020450  105777  160424            1$:     TSTB    @RSCS1          ;TEST
 4119  020454  100404                            BMI     2$              ;FOR READY TO COME BACK
 4120  020456  005367  160522                    DEC     REPT            ;WAIT
 4121  020462  001372                            BNE     1$              ;
 4122  020464  104000                            HLT                     ;READY NEVER CAME BACK
 4123  020466  005777  160406            2$:     TST     @RSCS1          ;ANY ERRORS?
 4124  020472  100032                            BPL     MRVRR           ;NO
 4125  020474  104050                            HLT     !DS!WC          ;STOP HERE WC FAILED
 4126                                                                    ;GO TO DZRSB DIAG
 4127                                                                    ;BEFORE TRYING TO DEBUG
 4128                                                                    ;THIS TEST
 4129  020476                    TBDIA:
 4130  020476  104402  020502                    TYPE    .+2             ;.ASCIZ <15><12>"FAILED VERIFY TEST --- RUN DZRSB DIAGNO
 4131  020560  000137  011466            MRVRR:  JMP     @#MRWRT         ;GO WRITE IN MAINTENANCE MODE
 4132                                    ;NOW CHECK TO SEE IF DRIVE WAS WRITTEN ON IN MAINTENANCE MODE
 4133
 4134  020564  104414                    MRVR2:  CLRDK                   ;CLEAR ALL REGISTERS
 4135  020566  012767  177777  160422            MOV     #177777,WORK    ;STALL
 4136  020574  005367  160416            3$:     DEC     WORK            ;WAITING FOR
 4137  020600  001375                            BNE     3$              ;DRIVE TO GET IN SYNC WITH INDEX PULSE
 4138  020602  012777  170000  160274            MOV     #-10000,@RSWC   ;SETUP WC FOR 1 TRACK
 4139  020610  052777  000010  160264            BIS     #BAI,@RSCS2     ;SET BAI
 4140  020616  012777  026666  160262            MOV     #INBUF,@RSBA    ;SETUP RSBA
 4141  020624  012767  177777  006034            MOV     #177777,INBUF   ;SETUP FOR COMPARE
 4142  020632  012777  000051  160240            MOV     #51,@RSCS1      ;DO A WRITE CHECK
 4143  020640  105777  160234            1$:     TSTB    @RSCS1          ;TEST FOR
 4144  020644  100375                            BPL     1$              ;READY TO COME BACK
 4145  020646  032777  040000  160226            BIT     #WCE,@RSCS2     ;DID WCE SET?
 4146  020654  001442                            BEQ     2$              ;NO
 4147  020656  104402  020662                    TYPE    .+2             ;.ASCIZ <15><12> "WRITE AMPLIFIER DID NOT GET DISABLED B
 4148  020756  104040                            HLT     !DS
 4149  020760  000404                            BR      4$              ;GET OUT
 4150  020762  005777  160112            2$:     TST     @RSCS1          ;ANY ERRORS?
 4151  020766  100001                            BPL     .+4             ;NO
 4152  020770  104040                            HLT     !DS             ;SHOULD NOT HAVE ANY ERRORS HERE
 4153  020772  000240                    4$:     NOP
 4154                                                                    ;TRY DZRSB DIAGNOSTIC
 4155
 4156  020774  052767  000001  160166  INFTST: BIS     #BIT0,ONCEE     ;SET FOUND DRIVE FLAG
 4157  021002  000137  002126                    JMP     @#TRYNX         ;GET NEXT DRIVE
```

```
4158                                    .SBTTL          $DONE - BELL AND SCOPE ROUTINE
4159
4160  021006  104400                    DONE:   SCOPE                           ;TERMINATIONG SCOPE FOR LOOPING
4161  021010  062767  000001  157770            ADD     #1,PCNT+2               ;ADD 1 TO THE PASS COUNT
4162  021016  005567  157762                    ADC     PCNT                    ;MAKE IT DOUBLE PREC.
4163  021022  032737  002000  177570            BIT     #SW10,@#SWR             ;RING THE BELL?
4164  021030  001004                            BNE     4$                      ;NO!
4165  021032  104402  021036                    TYPE    .+2                     ;.ASCIZ <BELL><177>
4166  021042  013700  000042            4$:     MOV     @#42,R0                 ;GET MONITOR ADDRESS
4167  021046  001404                            BEQ     3$                      ;IF NONE
4168  021050  004710                            JSR     7,(0)                   ;GO TO MONITOR
4169  021052  000240  000240  000240            240,240,240                     ;SAVE ROOM FOR ACT11
4170  021060  000167  000002            3$:     JMP     MULSYS                  ;RETURN
4171
4172  021064  000000                    .TBIT:  0                               ;T BIT FLAG
4173
4174                                    ;MULTI DRIVE SYSTEM?
4175
4176  021066                            MULSYS:
4177  021066  104402  021072                    TYPE    .+2                     ;.ASCIZ <15><12>"END OF PASS"
4178  021110  005067  157674                    CLR     LAD
4179  021114  005067  157660                    CLR     ICNT
4180  021120  032767  000020  160020            BIT     #BIT4,FLAG3             ;MULTI DRVIE?
4181  021126  001002                            BNE     1$                      ;NO
4182  021130  000137  001702                    JMP     @#MULTII                ;YES
4183  021134  000137  002332            1$:     JMP     @#NOWGO                 ;TEST ONLY ONE DRIVE
4184
4185                                    ;ERROR TYPEOUT ROUTINE FOR NO-OP TEST
4186
4187  021140  032767  000004  160022    NOPERR: BIT     #BIT2,ONCEE             ;WERE WE HERE BEFORE?
4188  021146  001031                            BNE     1$                      ;YES
4189  021150  052767  000004  160012            BIS     #BIT2,ONCEE             ;SET BEEN HERE BEFORE FLAG
4190  021156  104402  021162                    TYPE    .+2                     ;.ASCIZ <15><12>"ERROR CAUSED BY NO-OP FUNCTION "
4191  021224  016746  157766                    MOV     WORK,-(6)               ;PUT WORK ON STACK
4192  021230  104406                            TYPES                           ;TYPE STACK IN OCTAL - SUPRESS
4193  021232  000207                    1$:     RTS     PC
```

```
4194  021234  104402  021250          CHG:    TYPE    ,REGCHG         ;TYPE MESSAGE
4195  021240  016746  157752                  MOV     WORK,-(6)       ;PUT WORK ON STACK
4196  021244  104406                         TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
4197  021246  000207                         RTS     PC
4198
4199  021250  044103  047101  042507  REGCHG: .ASCIZ  "CHANGED WITH NO-OP FUNCTION "
4200  021256  020104  044527  044124
4201  021264  047040  026517  050117
4202  021272  043040  047125  052103
4203  021300  047511  020116     000
4204
4205  021305     015  051012  051115  TRMR:   .ASCIZ  <15><12>"RMR  DID NOT SET BY WRITING INTO "
4206  021312  020040  044504  020104
4207  021320  047516  020124  042523
4208  021326  020124  054502  053440
4209  021334  044522  044524  043516
4210  021342  044440  052116  020117
4211  021350     000
4212          021352                  .EVEN
4213
4214  021352  104422                  .MRINT: MRCLK                   ;CLOCK THE MAINT REG WITH A 11 AND A 1
4215  021354  104422                          MRCLK                   ;SAME
4216  021356  000002                         RTI                     ;RETURN
4217
4218  021360  012777  000011  157536  .MRCLK: MOV     #11,@RSMR       ;CLOCK THE
4219  021366  012777  000001  157530          MOV     #1,@RSMR        ;MAINT REG
4220  021374  062767  000001  157576          ADD     #1,MCCNT+2      ;ADD 1 TO CLOCK COUNT
4221  021402  005567  157570                  ADC     MCCNT           ;MAKE DOUBLE PRECISION
4222  021406  000002                         RTI
4223
4224  021410  017700  157510          .MRCK:  MOV     @RSMR,BAD       ;GET THE CONTENTS OF RSMR
4225  021414  017601  000000                  MOV     @(SP),GOOD      ;GET THE CORRECT ANSWER
4226  021420  062716  000002                  ADD     #2,(SP)         ;UPDATE THE RETURN ADDRESS FOR AN ERROR
4227  021424  020100                         CMP     GOOD,BAD        ;IS THE MR REG CORRECT?
4228  021426  001002                         BNE     1$              ;NO EXIT
4229  021430  062716  000002                  ADD     #2,(SP)         ;UPDATE RETURN ADDRESS TO SKIP THE HLT FOR CORRECT ANS
4230  021434  000002          1$:     RTI                             ;RETURN
4231
4232                          ;SEND INDEX PULSE TO THE MAINTENANCE REGISTER
4233  021436  012777  000021  157460  .MRIND: MOV     #21,@RSMR       ;SEND INDEX
4234  021444  012777  000001  157452          MOV     #1,@RSMR        ;PULSE TO MR REG
4235  021452  000002                         RTI
4236  021454  017700  157432          .DSCK:  MOV     @RSDS,BAD       ;GET THE CONTENTS OF RSDS
4237  021460  017601  000000                  MOV     @(SP),GOOD      ;GET THE CORRECT ANS
4238  021464  062716  000002                  ADD     #2,(SP)         ;UPDATE THE RETURN ADDR FOR AN ERROR
4239  021470  020100                         CMP     GOOD,BAD        ;IS RSDS CORRECT
4240  021472  001002                         BNE     1$              ;NO EXIT
4241  021474  062716  000002                  ADD     #2,(SP)         ;UPDATE RETURN ADDRTO SKIP THE HLT  FOR CORRRECT ANS
4242  021500  000002          1$:     RTI
```

```
                                        ;GET 1 BIT OF DATA FROM BUFFER
                                        ;SAVE THE LAST BIT TRANSFERED IN LOCATION LSTOD

021502 032767 000200 157434 .XBIT:  BIT   #BIT7,FLAG2     ;1ST 1 BIT OF 1ST WD?
021510 001446               BEQ   2$                      ;NO
021512 012767 000001 157432 MOV   #1,LSTOD                ;YES SETUP SYNC 1 BIT FOR END OF PREAMBLE;
                                                          ;TO CALCULATE BOTTOM BIT
021520 032767 000100 157442 BIT   #BIT6,ONCEE             ;1ST TIME THROUGH?
021526 001006               BNE   5$                      ;YES
021530 042767 000200 157406 BIC   #BIT7,FLAG2
021536 012767 000000 157406 MOV   #0,LSTOD
021544 042767 000100 157416 5$:   BIC   #BIT6,ONCEE       ;CLEAR 1ST TIME THROUGH FLAG
021552 005067 157432        4$:   CLR   CLKCNT            ;CLEAR CLOCK COUNTER AT START OF EACH WD
021556 032767 000400 157404 BIT   #BIT8,ONCEE             ;ON CRC WD?
021564 001062               BNE   1$                      ;YES
021566 005067 157364        CLR   NOWOD                   ;NO BITS 16 & 17 ARE 0

021572 032767 000400 157344 BIT   #BIT8,FLAG2             ;XFERING BIT 17?
021600 001003               BNE   7$                      ;YES
021602 042767 001000 157334 BIC   #BIT9,FLAG2             ;CLEAR FLAG FOR BIT 16
021610 042767 000400 157326 7$:   BIC   #BIT8,FLAG2       ;CLEAR FLAG FOR BIT 17
021616 012767 000020 157402 6$:   MOV   #16.,WORK3        ;LOOP 16 TIMES 1 FOR EACH BIT
021624 000002               RTI                           ;EXIT
021626 016767 157324 157316 2$:   MOV   NOWOD,LSTOD       ;SAVE LAST BIT XFERED
021634 032767 001000 157302 BIT   #BIT9,FLAG2
021642 001343               BNE   4$
021644 005767 157356        TST   WORK3                   ;DONE WITH WD YET?
021650 001013               BNE   3$                      ;NO
021652 032767 002000 157264 BIT   #BIT10,FLAG2            ;ON BIT 16 OF CRC WD?
021660 001334               BNE   4$                      ;YES
021662 062705 000002        ADD   #2,R5                   ;UPDATE BUFFER WD
021666 011504               MOV   (R5),R4                 ;GET DATA WD
021670 052767 001400 157246 BIS   #1400,FLAG2             ;SET BITS 8 & 9 IN FLAG2
021676 000725               BR    4$
021700 005067 157252        3$:   CLR   NOWOD             ;CLEAR PRESENT BIT
021704 032767 001000 157232 BIT   #BIT9,FLAG2             ;DID WE XFER BITS 16 & 17 YET?
021712 001317               BNE   4$                      ;NO
021714 000241               CLC
021716 006104               ROL   R4                      ;GET NEXT DATA BIT
021720 006167 157232        ROL   NOWOD                   ;PUT IT INTO NOWOD
021724 005367 157276        DEC   WORK3                   ;KEEP COUNT OF BITS IN THE WORD
021730 000002               RTI                           ;EXIT

                                        ;CRC IS BEING WRITTEN. BITS 17 & 16 ARE DATA BITS
                                        ;BITS 0 & 1 ARE ALWAYS 0

021732 005067 157220        1$:   CLR   NOWOD             ;CLEAR PRESENT BIT
021736 006104               ROL   R4                      ;GET NEXT BIT
021740 006167 157212        ROL   NOWOD                   ;TO BE XFERED
021744 032767 002000 157172 BIT   #BIT10,FLAG2            ;DONE WITH BITS 16 & 17 YET?
021752 001321               BNE   6$                      ;YES
021754 052767 002000 157162 BIS   #BIT10,FLAG2            ;NO
021762 042767 001000 157154 BIC   #BIT9,FLAG2
021770 000002               RTI                           ;EXIT
```

```
                                ;CLOCK ROUTINE (1ST OF ONE) WHICH IS USED TO CLOCK ONE BIT OF
                                ;DATA TO THE DRIVE AT A TIME.  THIS ROUTINE ALSO CHECKS THE PREVIOUS
                                ;BITS THAT HAVE BEEN TRANSFERRED AND CALCULATES WHICH STATE
                                ;THE MWDB BIT (BIT 12 IN THE MR REG) SHOULD BE IN

021772  104446                  .CLKD1: MCLK1                   ;CLOCK MR REG WITH AN 11
021774  005003                          CLR     R3              ;CLEAR WORK LOCATION
021776  005767  157154                  TST     NOWOD           ;TEST ODD BIT NOW BEING SENT FOR A 1 OR A 0
022002  001005                          BNE     TSTEVB          ;NOW TEST EVEN DATA BIT ON 1ST CLOCK
                                                                ;NOW BIT IS A 1 MWDB IS 0
022004  005767  157142          1$:     TST     LSTOD           ;TEST THE LAST ODD DATA BIT THAT WAS SENT
022010  001002                          BNE     TSTEVB          ;LAST ODD DATA BIT WAS A 1
                                                                ;MWDB IS A 0
022012  052703  010000          2$:     BIS     #BIT12,R3       ;SET MWDB FOR LATER COMPARE WITH MR REG

022016  012701  123511          TSTEVB: MOV     #123511,GOOD    ;GET CORRECT ANS
022022  050301                          BIS     R3,GOOD         ;FOR MR REG
022024  004767  001306                  JSR     PC,MRCAL        ;DETERMINE STATE OF SB & LSR BITS
022030  017700  157070                  MOV     @RSMR,BAD       ;GET CONTENTS OF MR REG
022034  020100                          CMP     GOOD,BAD        ;IS MR REG CORRECT?
022036  001002                          BNE     2$              ;NO TYPE OUT MR REG
022040  062716  000002                  ADD     #2,(SP)         ;UPDATE RETURN ADDR FOR CORRECT ANS
022044  000002          2$:     RTI                             ;RETURN
```

```
                                        ;SECOND CLOCK ROUTINE WHICH WILL FINISH TRANSFERRING THE DATA BIT
                                        ;THIS ROUTINE WILL CALCULATE WHAT MWDB SHOULD EQUAL IN THE
                                        ;MAINTENANCE REGISTER

022046  104450                  .CLKDO: MCLKO                   ;CLOCK MR REG
022050  005767  157102                  TST     NOWOD           ;IS THE PRESENT DATA BIT A 1?
022054  001403                          BEQ     1$              ;NO IT IS A 0
022056  052703  010000                  BIS     #BIT12,R3       ;SET MWDB FOR BIT BEING SENT IS A 1
022062  000402                          BR      4$
022064  042703  010000          1$:     BIC     #BIT12,R3       ;CLEAR MWDB FOR PRESENT BIT IS A 0
022070  012701  023501          4$:     MOV     #23501,GOOD     ;GET CORRECT ANS
022074  050301                          BIS     R3,GOOD         ;FOR MR REG
022076  004767  001234                  JSR     PC,MRCAL        ;DETERMINE STATE OF SB & LSR BITS
022102  017700  157016                  MOV     @RSMR,BAD       ;GET CONTENTS OF MR REG
022106  020100                          CMP     GOOD,BAD        ;IS MR REG CORRECT?
022110  001002                          BNE     5$              ;NO TYPEOUT ERROR
022112  062716  000002                  ADD     #2,(SP)         ;UPDATE ADDR FOR CORRECT ANS
022116  000002                  5$:     RTI                     ;RETURN

                                        ;TYPEOUT ROUTINE TO DETERMINE WHICH IC FAILED IN CRC TEST2
                                        ;AND TO TYPE IT OUT

022120  012767  022230  157070  CRCTYP: MOV     #CRCTAB,WORK    ;GET STARTING LOC OF IC TABLE
022126  012767  000001  157066          MOV     #1,WORK1        ;SETUP TO TEST FIRST CHIP
022134  036767  157062  157032  1$:     BIT     WORK1,SAVEE     ;WAS IT THIS BIT?
022142  001006                          BNE     2$              ;YES TYPE IT
022144  062767  000006  157044          ADD     #6,WORK         ;NO INDEX TABLE POINTER
022152  006167  157044                  ROL     WORK1           ;SETUP TO TEST NEXT CHIP
022156  000766                          BR      1$              ;NOW TES IT
022160  004777  157032          2$:     JSR     PC,@WORK        ;TYPE OUT CHIP
022164  104402  022170                  TYPE    ,.+2            ;.ASCIZ " IN THE CRC REG SHOULD BE SET"
022226  000207                          RTS     PC
```

```
4353                                        ;TABLE FOR CRC TEST 2 TYPEOUT ROUTINE
4354
4355  022230  104402  022420        CRCTAB: TYPE    ,E302
4356  022234  000207                        RTS     PC
4357  022236  104402  022426                TYPE    ,E305
4358  022242  000207                        RTS     PC
4359  022244  104402  022434                TYPE    ,E307
4360  022250  000207                        RTS     PC
4361  022252  104402  022442                TYPE    ,E3010
4362  022256  000207                        RTS     PC
4363  022260  104402  022451                TYPE    ,E3012
4364  022264  000207                        RTS     PC
4365  022266  104402  022460                TYPE    ,E3015
4366  022272  000207                        RTS     PC
4367  022274  104402  022467                TYPE    ,E242
4368  022300  000207                        RTS     PC
4369  022302  104402  022475                TYPE    ,E245
4370  022306  000207                        RTS     PC
4371  022310  104402  022503                TYPE    ,E247
4372  022314  000207                        RTS     PC
4373  022316  104402  022511                TYPE    ,E2410
4374  022322  000207                        RTS     PC
4375  022324  104402  022520                TYPE    ,E2412
4376  022330  000207                        RTS     PC
4377  022332  104402  022527                TYPE    ,E2415
4378  022336  000207                        RTS     PC
4379  022340  104402  022536                TYPE    ,E192
4380  022344  000207                        RTS     PC
4381  022346  104402  022544                TYPE    ,E197
4382  022352  000207                        RTS     PC
4383  022354  104402  022552                TYPE    ,E1910
4384  022360  000207                        RTS     PC
4385  022362  104402  022561                TYPE    ,E1915
4386  022366  000207                        RTS     PC
4387
4388                                        ;CLOCK MR REG WITH A 0-1
4389
4390  022370  012777  000001  156526  .MCLKB: MOV     #1,@RSMR
4391  022376  012777  000011  156520          MOV     #11,@RSMR
4392  022404  062767  000001  156566          ADD     #1,MCCNT+2
4393  022412  005567  156560                  ADC     MCCNT
4394  022416  000002                        RTI
```

```
 4395  022420  031505  026460  000062  E302:    .ASCIZ  "E30-2"
 4396  022426  031505  026460  000065  E305:    .ASCIZ  "E30-5"
 4397  022434  031505  026460  000067  E307:    .ASCIZ  "E30-7"
 4398  022442  031505  026460  030061  E3010:   .ASCIZ  "E30-10"
 4399  022450     000
 4400  022451     105  030063  030455  E3012:   .ASCIZ  "E30-12"
 4401  022456  000062
 4402  022460  031505  026460  032461  E3015:   .ASCIZ  "E30-15"
 4403  022466     000
 4404  022467     105  032062  031055  E242:    .ASCIZ  "E24-2"
 4405  022474     000
 4406  022475     105  032062  032455  E245:    .ASCIZ  "E24-5"
 4407  022502     000
 4408  022503     105  032062  033455  E247:    .ASCIZ  "E24-7"
 4409  022510     000
 4410  022511     105  032062  030455  E2410:   .ASCIZ  "E24-10"
 4411  022516  000060
 4412  022520  031105  026464  031061  E2412:   .ASCIZ  "E24-12"
 4413  022526     000
 4414  022527     105  032062  030455  E2415:   .ASCIZ  "E24-15"
 4415  022534  000065
 4416  022536  030505  026471  000062  E192:    .ASCIZ  "E19-2"
 4417  022544  030505  026471  000067  E197:    .ASCIZ  "E19-7"
 4418  022552  030505  026471  030061  E1910:   .ASCIZ  "E19-10"
 4419  022560     000
 4420  022561     105  034461  030455  E1915:   .ASCIZ  "E19-15"
 4421  022566  000065
 4422
 4423                                  ;CLOCK MR REG WITH A 1
 4424
 4425  022570  012777  000011  156326  .MCLK1:  MOV     #11,@RSMR
 4426  022576  062767  000001  156374           ADD     #1,MCCNT+2
 4427  022604  005567  156366                   ADC     MCCNT
 4428  022610  000002                           RTI
 4429
 4430                                  ;CLOCK MR REG WITH A0
 4431
 4432  022612  012777  000001  156304  .MCLK0:  MOV     #1,@RSMR
 4433  022620  000002                           RTI
```

```
                                        ;GET ONE BIT OF DATA FROM INBUF
                                        ;FOR READING FROM DRIVE TO DETERMINE THE
                                        ;STATE OF MRD8 IN THE MR REG.
022622 005767 156400           .RBIT:   TST     WORK3          ;STARTING NEW WD?
022626 001035                           BNE     3$             ;NO
022630 062705 000002                    ADD     #2,R5          ;UPDATE BUFFER WD
022634 011504                           MOV     (R5),R4        ;GET DATA WD
022636 052767 004000 156300             BIS     #BIT11,FLAG2   ;SET TO INDICATE BIT 17
022644 005067 156340           5$:      CLR     CLKCNT         ;CLEAR CLOCK COUNTER AT START OF EACH WD
022650 032767 000400 156312             BIT     #BIT8,ONCEE    ;ON CRC WD?
022656 001041                           BNE     1$             ;YES
022660 032767 000040 156256             BIT     #BIT5,FLAG2    ;IN CRC TEST ????
022666 001407                           BEQ     7$             ;NO
022670 012767 000020 156330             MOV     #16.,WORK3
022676 042767 004000 156240             BIC     #BIT11,FLAG2   ;FOR CRC TEST
022704 000416                           BR      4$
022706 005067 156244           7$:      CLR     NOWOD          ;BITS 16 + 17 OF DATA WORD ARE 0
022712 012767 000020 156306    6$:      MOV     #16.,WORK3     ;16 LOOPS FOR REMAINING 16 BITS OF WORD
022720 000002                           RTI
022722 032767 004000 156214    3$:      BIT     #BIT11,FLAG2   ;IS THIS BIT 16?
022730 001404                           BEQ     4$             ;NO
022732 042767 004000 156204             BIC     #BIT11,FLAG2   ;RTRANSFER BIT 16
022740 000741                           BR      5$
022742 005067 156210           4$:      CLR     NOWOD          ;CLEAR PRESENT BIT
022746 006104                           ROL     R4             ;GET NEXT DATA BIT
022750 006167 156202                    ROL     NOWOD          ;SAVE IT IN ODD BIT
022754 005367 156246                    DEC     WORK3          ;KEEP COUNT OF BITS IN THE WORD
022760 000002                           RTI                    ;RETURN
                                        ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS, 0 & 1 ARE ALWAYS 0
022762 005067 156170           1$:      CLR     NOWOD          ;GET BITS 17
022766 006104                           ROL     R4             ;AND 16
022770 006167 156162                    ROL     NOWOD          ;FOR CRC WORD
022774 000746                           BR      6$             ;CONTINUE
```

```
 4468  022776  004767  000312          .CLKR1: JSR     PC,CALRTB       ;CALCULATE MRDB BIT FOR MR REG
 4469  023002  012703  000011                  MOV     #11,R3          ;SETUP CLOCK BITS
 4470  023006  062767  000001  156164          ADD     #1,MCCNT+2      ;INCREMENT
 4471  023014  005567  156156                  ADC     MCCNT           ;CLOCK COUNT
 4472  023020  056703  156172                  BIS     WORK,R3         ;SET BOTTOM BITS
 4473  023024  010377  156074                  MOV     R3,@RSMR        ;SEND
 4474  023030  012701  133611                  MOV     #133611,GOOD    ;CALCULATE CORRECT ANS FOR MR REG
 4475  023034  032767  000004  156102          BIT     #BIT2,FLAG2     ;WRITE CK TEST?
 4476  023042  001402                          BEQ     7$              ;NO
 4477  023044  052701  000100                  BIS     #BIT6,GOOD      ;YES SET RD IN MR REG
 4478  023050  050301                  7$:     BIS     R3,GOOD
 4479  023052  032767  000400  156110          BIT     #BIT8,ONCEE     ;ON CRC WD?
 4480  023060  001406                          BEQ     2$              ;NO
 4481  023062  022767  000022  156114          CMP     #22,REPT        ;SHOULD CRCW BE SET?
 4482  023070  001402                          BEQ     2$              ;YES
 4483  023072  042701  020000                  BIC     #20000,GOOD     ;CLEAR CRCW
 4484  023076  005367  156104          2$:     DEC     REPT1           ;SHOULD SB SET
 4485  023102  001017                          BNE     6$              ;NO
 4486  023104  012767  000044  156074          MOV     #44,REPT1       ;RESET SB COUNTER
 4487  023112  052701  004000                  BIS     #BIT11,GOOD     ;SET SB
 4488  023116  032767  000400  156044  3$:     BIT     #BIT8,ONCEE     ;ON CRC WD?
 4489  023124  001406                          BEQ     6$              ;NO
 4490  023126  022767  000043  156052          CMP     #43,REPT1       ;SHOULD SB AND CRCW BE SET ?
 4491  023134  001002                          BNE     6$              ;NO
 4492  023136  052701  020000                  BIS     #20000,GOOD     ;SET SB AND CRCW
 4493  023142  017700  155756          6$:     MOV     @RSMR,BAD       ;GET MR REG
 4494  023146  020100                          CMP     GOOD,BAD        ;IS RSMR CORRECT?
 4495  023150  001002                          BNE     4$              ;NO
 4496  023152  062716  000002                  ADD     #2,(SP)         ;YES
 4497  023156  000002                  4$:     RTI                     ;RETURN
```

```
 4498  023160  056703  156032          .CLKRD: BIS     WORK,R3         ;SET BOTTOM BITS
 4499  023164  042703  000010                  BIC     #BIT3,R3
 4500  023170  010377  155730                  MOV     R3,@RSMR        ;SEND
 4501  023174  012701  023601                  MOV     #23601,GOOD     ;CALCULATE CORRECT ANS FOR MR REG
 4502  023200  032767  000004  155736          BIT     #BIT2,FLAG2     ;WRITE CK TEST?
 4503  023206  001402                          BEQ     7$              ;NO
 4504  023210  052701  000100                  BIS     #BIT6,GOOD      ;YES SET RD IN MR REG
 4505  023214  050301                  7$:     BIS     R3,GOOD
 4506  023216  032767  000400  155744          BIT     #BIT8,ONCEE     ;ON CRC WD?
 4507  023224  001402                          BEQ     2$              ;NO
 4508  023226  042701  020000                  BIC     #20000,GOOD     ;CLEAR CRCW
 4509  023232  005367  155750          2$:     DEC     REPT1           ;SHOULD SB SET?
 4510  023236  001017                          BNE     6$              ;NO
 4511  023240  012767  000022  155740          MOV     #18.,REPT1      ;RESET SB COUNTER
 4512  023246  052701  004000                  BIS     #BIT11,GOOD     ;SET SB
 4513  023252  032767  000400  155710  3$:     BIT     #BIT8,ONCEE     ;ON CRC WD?
 4514  023260  001406                          BEQ     6$              ;NO
 4515  023262  022767  000022  155716          CMP     #22,REPT1       ;SHOULD SB AND CRCW BE SET ?
 4516  023270  001002                          BNE     6$              ;NO
 4517  023272  052701  020000                  BIS     #20000,GOOD     ;SET SB AND CRCW
 4518  023276  017700  155622          6$:     MOV     @RSMR,BAD       ;GET MR REG
 4519  023302  020100                          CMP     GOOD,BAD        ;IS RSMR CORRECT?
 4520  023304  001002                          BNE     4$              ;NO
 4521  023306  062716  000002                  ADD     #2,(SP)         ;YES
 4522  023312  000002                  4$:     RTI                     ;RETURN
```

```
4523                                    ;CALCULATE THE STATE OF MRDB FROM CURRENT INPUT BIT
4524                                    ;LOCATION WORK CONTAINS CORRECT DATA FOR  MRDB
4525  023314  005067  155676            CALRTB: CLR     WORK            ;CLEAR WORK LOCATION
4526  023320  005767  155632                    TST     NOWOD           ;IS CURRENT BIT A 0?
4527  023324  001403                            BEQ     2$              ;YES
4528  023326  052767  000004  155662            BIS     #BIT2,WORK      ;NO SET MRDB
4529  023334  000207                    2$:     RTS     PC              ;RETURN
4530
4531                                    ;CALCULATE MR REG TO DETERMINE THE STATE OF THE CRC-SB AND LSR BITS
4532                                    ;ON THE DIFFERENT CLOCKS ON THE DIFFERENT WORDS THROUGHOUT THE SECTOR
4533
4534  023336  005267  155646            MRCAL:  INC     CLKCNT          ;ADD ONE TO CLOCK COUNT OF WORD
4535  023342  032767  000200  155620            BIT     #BIT7,ONCEE     ;TRANSFERRING LAST WORD?
4536  023350  001032                            BNE     LSTWD           ;YES
4537  023352  032767  000400  155610            BIT     #BIT8,ONCEE     ;TRANSFERRING CRC WORD?
4538  023360  001051                            BNE     CRCWD           ;YES
4539  023362  022767  000016  155620            CMP     #16,CLKCNT      ;CLOCK COUNT 16 OR GREATER?
4540  023370  101401                            BLOS    1$              ;YES
4541  023372  000406                            BR      2$              ;GET OUT
4542  023374  022767  000040  155606    1$:     CMP     #40,CLKCNT      ;CLOCK COUNT 40 OR GREATER?
4543  023402  101402                            BLOS    2$              ;YES GET OUT
4544  023404  052701  004000                    BIS     #BIT11,GOOD     ;SET SB BIT
4545  023410  022767  000037  155572    2$:     CMP     #37,CLKCNT
4546  023416  001404                            BEQ     3$
4547  023420  022767  000040  155562            CMP     #40,CLKCNT
4548  023426  001002                            BNE     4$
4549  023430  042701  002000            3$:     BIC     #BIT10,GOOD     ;CLEAR LSR
4550  023434  000207                    4$:     RTS     PC              ;RETURN
4551
4552                                    ;CALCULATE MR FOR LAST DATA WORD
4553  023436  022767  000036  155544    LSTWD:  CMP     #36,CLKCNT      ;IS THIS CLOCK 36 OR LESS?
4554  023444  103016                            BHIS    2$              ;YES GETOUT
4555  023446  022767  000037  155534            CMP     #37,CLKCNT      ;IS THIS CLOCK 15?
4556  023454  001003                            BNE     3$              ;NO
4557  023456  042701  002000            4$:     BIC     #BIT10,GOOD     ;YES CLEAR LSR
4558  023462  000407                            BR      2$
4559  023464  022767  000040  155516    3$:     CMP     #40,CLKCNT
4560  023472  001001                            BNE     5$
4561  023474  000770                            BR      4$
4562  023476  042701  020000            5$:     BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
4563  023502  000207                    2$:     RTS     PC
4564
4565                                    ;CALCULATE MR FOR CRC WORD
4566
4567  023504  042701  020000            CRCWD:  BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
4568  023510  022767  000037  155472            CMP     #37,CLKCNT      ;IS THIS CLOCK 17?
4569  023516  001002                            BNE     2$              ;NO
4570  023520  042701  002000                    BIC     #BIT10,GOOD     ;CLEAR LSR BIT
4571  023524  022767  000040  155456    2$:     CMP     #40,CLKCNT
4572  023532  001002                            BNE     1$
4573  023534  042701  002000                    BIC     #BIT10,GOOD
4574  023540  000207                    1$:     RTS     PC              ;RETURN
```

```
                                        ;GENERATE A CRC WORD FROM THE DATA BUFFER
                                        ;AND LEAVE THE CRC WORD IN "WORK" LOCATION
                                        ;EXIT ROUTINE WITH RTS PC

023542  012767  000100  155434  GENCRC: MOV     #64.,REPT       ;64 WORDS PER SECTOR
023550  032767  000040  155366          BIT     #BIT5,FLAG2     ;IN CRC TEST?
023556  001403                        BEQ     13$             ;NO
023560  012767  000110  155416          MOV     #72.,REPT       ;YES
023566  012705  026666          13$:    MOV     #INBUF,R5       ;GET STARTING ADDR OF OUTPUT BUFFER
023572  011504                        MOV     (R5),R4         ;GET DATA WD
023574  005067  155420                CLR     WORK0           ;CLEAR WORK LOCATION

                                        ;INBIT CONTAINS PRESENT INPUT BIT
                                        ;WK15 = BIT15 IF CRC AT TIME T
                                        ;WORK0 = CRC AT TIME T + DURING FINAL MANIPULATION
                                        ;WORK = BITS FROM SAVED CRC WORD (WCRC)

023600  012767  000022  155400  1$:     MOV     #18.,REPT1      ;GET 18 BITS PER WD
023606  032767  000040  155330          BIT     #BIT5,FLAG2     ;IN CRC TEST?
023614  001403                        BEQ     2$              ;NO
023616  012767  000020  155362          MOV     #16.,REPT1      ;YES
023624  016767  155370  155350  2$:     MOV     WORK0,WCRC      ;SAVE CURRENT CRC WD
023632  005067  155356                CLR     WK15            ;CLEAR BIT 15 FROM CRC AT T 1
023636  000241                        CLC                     ;CLEAR CARY
023640  006167  155354                ROL     WORK0           ;SHIFT CRC WD LEFT
023644  006167  155344                ROL     WK15            ;CONTAINS BIT 15 OF CRC
023650  032767  000040  155266          BIT     #BIT5,FLAG2     ;IN CRC TEST?
023656  001004                        BNE     12$             ;YES
023660  022767  000021  155320          CMP     #17.,REPT1      ;DONE BITS 16 AND 17 YET?
023666  101406                        BLOS    3$              ;NO
023670  005067  155316          12$:    CLR     INBIT           ;CLEAR WORK LOC
023674  006104                        ROL     R4              ;PUT DATA BIT FROM BUFFER
023676  006167  155310                ROL     INBIT           ;IN WORK1 LOC
023702  000402                        BR      4$
023704  005067  155302          3$:     CLR     INBIT           ;FOR BITS 16 AND 17
023710  016767  155300  155300  4$:     MOV     WK15,WORK       ;GET BIT 15 OF CRC
023716  004767  000220          5$:     JSR     PC,XXOR         ;XOR BIT15 WITH INPUT BIT
023722  042767  000001  155270          BIC     #BIT0,WORK0
023730  005767  155256                TST     INBIT           ;TEST RESULT OF XOR
023734  001403                        BEQ     6$
023736  052767  000001  155254          BIS     #BIT0,WORK0
023744  016767  155242  155206  6$:     MOV     INBIT,RS0       ;SAVE XOR SESULT OF BIT 0 AND INPUT
```

```
4618                                        ;FROM B0 IN WORK0 AND B1 IN SAVED CRC (WCRC) CLACULATE
4619                                        ;NEW B2 FOR WORK0
4620
4621  023752  005067  155240                      CLR     WORK
4622  023756  032767  000002  155216              BIT     #BIT1,WCRC
4623  023764  001403                              BEQ     7$
4624  023766  052767  000001  155222              BIS     #BIT0,WORK
4625  023774  016767  155160  155210  7$:         MOV     RS0,INBIT
4626  024002  004767  000134                      JSR     PC,XXOR
4627  024006  042767  000004  155204              BIC     #BIT2,WORK0
4628  024014  005767  155172                      TST     INBIT                   ;TEST RESULT OF XOR
4629  024020  001403                              BEQ     8$
4630  024022  052767  000004  155170              BIS     #BIT2,WORK0
4631
4632                                        ;FROM B0 IN WORK0 AND B14 IN WCRC CLACULATE BIT15 IN WORK0
4633
4634  024030  005067  155162                8$:   CLR     WORK
4635  024034  032767  040000  155140              BIT     #BIT14,WCRC
4636  024042  001403                              BEQ     9$
4637  024044  052767  000001  155144              BIS     #BIT0,WORK
4638  024052  016767  155102  155132  9$:         MOV     RS0,INBIT
4639  024060  004767  000056                      JSR     PC,XXOR
4640  024064  042767  100000  155126              BIC     #BIT15,WORK0
4641  024072  005767  155114                      TST     INBIT                   ;TEST RESULT OF XOR
4642  024076  001403                              BEQ     10$
4643  024100  052767  100000  155112              BIS     #BIT15,WORK0
4644  024106  005367  155074                10$:  DEC     REPT1                   ;DONE WITH WD
4645  024112  001244                              BNE     2$                      ;NO
4646  024114  005367  155064                      DEC     REPT                    ;DONE WITH SECTOR?
4647  024120  001404                              BEQ     11$                     ;YES
4648  024122  062705  000002                      ADD     #2,R5                   ;GET NEXT WD
4649  024126  011504                              MOV     (R5),R4                 ;GET DATA WD
4650  024130  000623                              BR      1$
4651  024132  016767  155062  155056  11$:        MOV     WORK0,WORK              ;SAVE CRC WORD IN WORK
4652  024140  000207                              RTS     PC                      ;EXIT
4653
4654                                        ;XOR SUBROUTINE
4655
4656  024142  016703  155050                XXOR: MOV     WORK,R3
4657  024146  046703  155040                      BIC     INBIT,R3
4658  024152  046767  155040  155032              BIC     WORK,INBIT
4659  024160  050367  155026                      BIS     R3,INBIT
4660  024164  000207                              RTS     PC
```

```
                                .SBTTL          $TYPE - TTY TYPEOUT ROUTINE

                                ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
                                ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
                                ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
                                ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
                                ;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
                                ;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
                                ;IS IN FILCHR+1.

024166  010446                .TYPE:  MOV     R4,-(6)         ;SAVE R4
024170  010546                        MOV     R5,-(6)         ;SAVE R5
024172  017605  000004                MOV     @4(6),R5        ;GET ADDRESS TO BE TYPED
024176  032705  177400                BIT     #177400,R5      ;IS IT A TYPEM?
024202  001002                        BNE     1$              ;NO
024204  016605  000004                MOV     4(6),R5         ;GET ADDRESS OF CHARACTER
024210  105715                1$:     TSTB    (R5)            ;TERMINATOR?
024212  001423                        BEQ     2$              ;GET OUT IF SO
024214  122715  000012                CMPB    #12,(R5)        ;IS THE CHAR A LINE FEED
024220  001012                        BNE     4$              ;NO - GET OUT
024222  116704  154567                MOVB    FILCHR+1,R4     ;GET THE FILL COUNT
024226  116777  154562  154564  5$:   MOVB    FILCHR,@TPB     ;TYPE A FILLER
024234  105777  154556                TSTB    @TPS            ;DONE YET?
024240  100375                        BPL     .-4             ;NO - WAIT
024242  005304                        DEC     R4              ;DEC COUNT
024244  001370                        BNE     5$              ;LOOP UNTIL 0
024246  112577  154546        4$:     MOVB    (R5)+,@TPB      ;LOAD AND TYPE THE CHARACTER
024252  105777  154540                TSTB    @TPS            ;IS THE PRINTER READY
024256  100375                        BPL     .-4             ;WAIT UNTIL IT IS
024260  000753                        BR      1$              ;GET THE NEXT CHARACTER
024262  017646  000004        2$:     MOV     @4(6),-(6)      ;GET ADDRESS TO BE TYPED
024266  062766  000002  000006        ADD     #2,6(6)         ;ADD 2 TO THE ADDRESS
024274  022666  000004                CMP     (6)+,4(6)       ;IS IT .+2?
024300  001006                        BNE     3$              ;NO
024302  062705  000002                ADD     #2,R5           ;ADD 2 TO THE ADDRESS
024306  042705  000001                BIC     #1,R5           ;BACK UP TO AN EVEN BYTE
024312  010566  000004                MOV     R5,4(6)         ;RESTORE ADDRESS
024316  012605                3$:     MOV     (6)+,R5         ;RESTORE R5
024320  012604                        MOV     (6)+,R4         ;RESTORE R4
024322  000002                        RTI                     ;RETURN
```

```
                                .SBTTL          $SCOPE - SCOPE LOOP HANDLER

                                ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
                                ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
                                ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
                                ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"

024324  032737  000400  177570  .SCOPE: BIT     #SW8,@#SWR      ;LOOP ON SPEC. TEST?
024332  001404                          BEQ     1$              ;NO LOOP ON SPEC. TEST
024334  123767  177570  154436          CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   #SW7-0#
024342  001453                          BEQ     .OVER           ;NOT RIGHT TEST
024344  032737  040000  177570  1$:     BIT     #SW14,@#SWR     ;LOOP ON TEST?
024352  001045                          BNE     .KIT            ;LOOP ON TEST IS SET
024354  000416                          BR      3$              ;SKIP - NOP FOR XOR TESTER
024356  013746  000004                  MOV     @#4,-(6)        ;PUSH @#4 ON STACK
024362  012737  024402  000004          MOV     #4$,@#4         ;SET FOR TIMEOUT
024370  005737  177060                  TST     @#177060        ;ERROR ON XOR?
024374  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
024400  000422                          BR      .SVLAD          ;NO ERROR - GO TO NEXT TEST
024402  022626                  4$:     CMP     (6)+,(6)+       ;CLEAR STACK
024404  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
024410  000426                          BR      .KIT            ;ERROR - LOOP ON TEST
024412  032737  004000  177570  3$:     BIT     #SW11,@#SWR     ;KILL ITERATIONS
024420  001012                          BNE     .SVLAD          ;YES - KILL ITERATIONS
024422  105767  154353                  TSTB    ICNT+1          ;FIRST ONE?
024426  001404                          BEQ     2$              ;BRANCH IF FIRST
024430  126767  000060  154343          CMPB    TIMES,ICNT+1    ;DONE?
024436  003013                          BGT     .KIT            ;BRANCH IF NOT
024440  112767  000001  154333  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
024446  105267  154326          .SVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
024452  011667  154332                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
024456  016737  154316  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
024464  000002                          RTI                     ;RETURN

024466  105267  154307          .KIT:   INCB    ICNT+1          ;INC THE ITERATION COUNT
024472  016737  154302  177570  .OVER:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
024500  005767  154304                  TST     LAD             ;FIRST ONE?
024504  001760                          BEQ     .SVLAD          ;YES
024506  016716  154276                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
024512  000002                          RTI                     ;FIXES PS

024514  000001          TIMES:  1               ;RUN 1 TIMES
```

```
                                        .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)

                                        ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
                                        ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
                                        ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
                                        ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
                                        ;(HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADITIONAL TYPEOUTS.

024516  032737  002000  177570  .HLT:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
024524  001402                          BEQ     1$              ;NO - SKIP
024526  104402  000007                  TYPE    ,BELL           ;RING BELL
024532  005267  154244          1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
024536  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
024544  001025                          BNE     2$              ;SKIP TYPEOUTS
024546  104402  024552                  TYPE    .+2             ;.ASCIZ <15><12>
024556  011667  154230                  MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
024562  162767  000002  154222          SUB     #2,HLTADR       ;FUDGE ADDRESS
024570  117767  154216  000054          MOVB    @HLTADR,.HLTCT  ;GET HLT ARGUEMENT
024576  016746  154210                  MOV     HLTADR,-(6)     ;PUT HLTADR ON STACK
024602  104404                          TYPEO                   ;TYPE STACK IN OCTAL
024604  104402  024610                  TYPE    .+2             ;.ASCIZ "  "
024614  004767  001146                  JSR     PC,RSREG        ;GO TO USER ERROR ROUTINE
024620  005737  177570          2$:     TST     @#SWR           ;HALT ON ERROR
024624  100001                          BPL     .+4             ;SKIP IF CONTINUE
024626  000000                          HALT                    ;HALT ON ERROR!
024630  032737  001000  177570          BIT     #SW9,@#SWR      ;CHECK FOR INHIBIT LOOP ON ERROR
024636  001003                          BNE     3$              ;SKIP IF LOOP ON ERROR
024640  105067  154135                  CLRB    ICNT+1          ;CLEAR ITERATION COUNT
024644  000002                          RTI                     ;RETURN
024646  000167  177614          3$:     JMP     .KIT            ;LOOP ON TEST UNTIL NO ERRORS

024652  000000                  .HLTCT: 0                       ;HLT ARGUMENT
```

```
                                        .SBTTL          $OCTAL - OCTAL TYPEOUT ROUTINE

                                        ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
                                        ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
                                        ;16 BITS. IT IS CALLED VIA THE TYPOCT, TYPBIT, OR TYPOCS MACRO'S.

024654  012767  170101  000160  .TYPEB: MOV     #170101,.PR     ;SET BIT FLAG AND 16. CHARACTER COUNT
024662  000411                  BR      .PTIT           ;NOW TYPE IT IN BIT FORM
024664  112767  000001  000150  .TYPEO: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
024672  000402                  BR      .+6             ;SKIP
024674  005067  000142          .TYPES: CLR     .PR             ;SUPRESS LEADING ZERO'S
024700  112767  177772  000135          MOVB    #-6,.PR+1       ;SET COUNT
024706                          .PTIT:
024706  010446                  MOV     R4,-(6)         ;PUSH R4 ON STACK
024710  010546                  MOV     R5,-(6)         ;PUSH R5 ON STACK
024712  016605  000010          MOV     10(6),R5        ;GET THE DATA
024716  012704  025044          MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
024722  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
024724  000411                  BR      .PRF            ;ROTATE FIRST BIT
024726  105014                  .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
024730  032767  000100  000104          BIT     #100,.PR        ;BIT TYPING MODE?
024736  001004                  BNE     .PRF            ;YES - SKIP 2 ROTATES
024740  006105                  ROL     R5              ;ROTATE BIT INTO C
024742  106114                  ROLB    (4)             ;PACK IT
024744  006105                  ROL     R5              ;ROTATE BIT INTO C
024746  106114                  ROLB    (4)             ;PACK IT
024750  006105                  .PRF:   ROL     R5              ;ROTATE BIT INTO C
024752  106114                  ROLB    (4)             ;PACK IT
024754  105714                  TSTB    (4)             ;IS IT ZERO?
024756  001402                  BEQ     .+6             ;SKIP INC
024760  105267  000056          INCB    .PR             ;SET FILL SWITCH
024764  105767  000052          TSTB    .PR             ;CHECK FILL SWITCH
024770  001402                  BEQ     .+6             ;SKIP BITSET
024772  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
024776  105267  000041          INCB    .PR+1           ;INC COUNT
025002  001351                  BNE     .PRL            ;REPEAT
025004  022704  025044          CMP     #.PR+2,R4       ;EMPTY BUFFER?
025010  001002                  BNE     .+6             ;SKIP IF NOT
025012  112724  000060          MOVB    #'0,(4)+        ;LOAD 1 ZERO
025016  105014                  CLRB    (4)             ;NULL TERMINATOR
025020  104402  025044          TYPE    .PR+2           ;TYPE IT
025024  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
025026  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
025030  016666  000002  000004  MOV     2(6),4(6)       ;GET RID OF
025036  012616                  MOV     (6)+,(6)        ;DATA WORD
025040  000002                  RTI                     ;RETURN

025042  000012          .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
                                .SBTTL          $POWER - POWER DOWN AND UP ROUTINES

                                ;THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
                                ;THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
                                ;WAIT FOR POWER TO GO DOWN AND BE RESTORED.
                                ;IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
                                ;THE PROGRAM WILL HALT AT '.ILLUP'.

025066  012777  025214  000126  .POWER: MOV     #.ILLUP,@.PUVEC ;SET FOR FAST UP
025074  012777  000340  000122          MOV     #340,@.PUVEC$+2 ;PRIO:7
025102  010046                  MOV     R0,-(6)         ;PUSH R0 ON STACK
025104  010146                  MOV     R1,-(6)         ;PUSH R1 ON STACK
025106  010246                  MOV     R2,-(6)         ;PUSH R2 ON STACK
025110  010346                  MOV     R3,-(6)         ;PUSH R3 ON STACK
025112  010446                  MOV     R4,-(6)         ;PUSH R4 ON STACK
025114  010546                  MOV     R5,-(6)         ;PUSH R5 ON STACK
025116  010667  000076          MOV     SP,.SAVR6       ;SAVE SP
025122  012777  025132  000072  MOV     #.POWUP,@.PUVEC ;SET UP VECTOR
025130  000000                  HALT                    ;WAIT FOR PF

025132  016706  000062  .POWUP: MOV     .SAVR6,SP       ;GET SP
025136  005001                  CLR     R1              ;WAIT LOOP FOR THE TTY
025140  005201          1$:     INC     R1              ;WAIT FOR THE INC
025142  001376                  BNE     1$              ;OF   WORD
025144  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
025146  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
025150  012603                  MOV     (6)+,R3         ;POP STACK INTO R3
025152  012602                  MOV     (6)+,R2         ;POP STACK INTO R2
025154  012601                  MOV     (6)+,R1         ;POP STACK INTO R1
025156  012600                  MOV     (6)+,R0         ;POP STACK INTO R0
025160  012737  025066  000024  MOV     #.POWER,@#24    ;SET UP THE POWER DOWN VECTOR
025166  012737  000340  000026  MOV     #340,@#26       ;PRIO:7
025174  104402  025200          TYPE    ,.+2            ;.ASCIZ <15><12>"POWER"
025210  000167  173652          JMP     MULSYS          ;JMP TO USER ADDRESS

025214  000000  .ILLUP: HALT                            ;THE POWER UP SEQUENCE WAS STARTED
025216  000776          BR      .-2                     ; BEFORE THE POWER DOWN WAS COMPLETE

025220  000000  .SAVR6: 0                               ;PUT THE SP HERE
025222  000024  000026  .PUVEC: 24,26                    ;POWER UP VECTOR
```

```
                                        .SBTTL          $RDOCT - OCTAL INPUT ROUTINE

                                        ;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
                                        ;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.

025226  011646                  .RDOCT: MOV     (6),-(6)        ;MOVE THE PC
025230  016666  000004  000002          MOV     4(6),2(6)       ;MOVE THE PS
025236  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
025240  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
025242  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
025244  104412                  4$:     RDLIN                   ;READ A LINE INTO INPUT
025246  005001                          CLR     R1              ;INIT DATA WORD
025250  012703  025450                  MOV     #INPUT,R3       ;INIT POINTER
025254  112302                  1$:     MOVB    (3)+,R2         ;GET A BYTE
025256  001417                          BEQ     2$              ;GET OUT IF ZERO
025260  122702  000060                  CMPB    #'0,R2          ;CHECK FOR 0 OR GREATER
025264  003022                          BGT     3$              ;ERROR - LESS THAN 0
025266  122702  000067                  CMPB    #'7,R2          ;CHECK FOR 7 OR LESS
025272  002417                          BLT     3$              ;ERROR - GREATER THAN 7
025274  006002                          ROR     R2              ;GET
025276  006002                          ROR     R2              ;INTO
025300  006002                          ROR     R2              ;POSITION
025302  006101                          ROL     R1              ;FIRST BIT
025304  006102                          ROL     R2              ;GET
025306  006101                          ROL     R1              ;SECOND BIT
025310  006102                          ROL     R2              ;GET
025312  006101                          ROL     R1              ;THIRD BIT
025314  000757                          BR      1$              ;LOOP
025316  010166  000012          2$:     MOV     R1,12(6)        ;SAVE THE RESULT
025322  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
025324  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
025326  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
025330  000002                          RTI                     ;RETURN

025332                          3$:
025332  104402  025336                  TYPE    .+2             ;.ASCIZ "?"<15><12>
025342  000740                          BR      4$              ;TRY AGAIN
```

```
                                        .SBTTL          $RDLIN - TTY INPUT ROUTINE
                                        ;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
                                        ;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
                                        ;INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYPING
                                        ;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RL'9OUT.

025344  010546                  .RDLIN: MOV     R5,-(6)         ;SAVE R5
025346  012705  025450          1$:     MOV     #INPUT,R5       ;GET ADDRESS
025352  022705  025470          2$:     CMP     #INPUT+16.,R5   ;BUFFER FULL?
025356  001412                          BEQ     4$              ;YES - TYPE "?"
025360  105737  177560                  TSTB    @#177560        ;WAIT FOR
025364  100375                          BPL     .-4             ;A CHARACTER
025366  113715  177562                  MOVB    @#177562,(5)    ;GET CHARACTER
025372  142715  000200                  BICB    #200,(5)        ;GET RID OF JUNK
025376  122715  000177                  CMPB    #177,(5)        ;IS IT A RUBOUT
025402  001005                          BNE     3$              ;SKIP IF NOT
025404                          4$:
025404  104402  025410                  TYPE    .+2             ;.ASCIZ "?"<15><12>
025414  000754                          BR      1$              ;ZAP THE BUFFER AND LOOP
025416  111527  000000          3$:     MOVB    (5),#0          ;SET UP FOR TYPING
025422  104402  025420                  TYPE    .3$+2           ;ECHO IT
025426  122725  000015                  CMPB    #15,(5)+        ;CHECK FOR RETURN
025432  001347                          BNE     2$              ;LOOP IF NOT RETURN
025434  105065  177777                  CLRB    -1(5)           ;ZAP RETURN (THE 15)
025440  104402  000012                  TYPE    .12             ;TYPE A LINE FEED
025444  012605                          MOV     (6)+,R5         ;RESTORE R5
025446  000002                          RTI                     ;RETURN

025450  000020          INPUT:  .BLKB   16.             ;TTY INPUT AREA
```

```
                                .SBTTL          $TRAP - TRAP HANDLER

                                ;THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
                                ;SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
                                ;THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
                                ;FOLLOW THIS MACRO.

025470  011646          .TRAP:  MOV     (6),-(6)        ;GET ADDRESS OF TRAP +2
025472  162716  000002          SUB     #2,(6)          ;MAKE IT ADDRESS OF TRAP
025476  017616  000000          MOV     @(6),(6)        ;GET TRAP INSTRUCTION
025502  062716  121110          ADD     #.TRP+2-TRAP,(6);GET DATA AND MAKE IT AN OFFSET
025506  013607          .TRP:   MOV     @(6)+,PC        ;GO TO PROPER SUBROUTINE

025510  024324                  .SCOPE                  ;SCOPE  =  TRAP+0       (104400)
025512  024166                  .TYPE                   ;TYPE   =  TRAP+2       (104402)
025514  024664                  .TYPEO                  ;TYPEO  =  TRAP+4       (104404)
025516  024674                  .TYPES                  ;TYPES  =  TRAP+6       (104406)
025520  025226                  .RDOCT                  ;RDOCT  =  TRAP+10      (104410)
025522  025344                  .RDLIN                  ;RDLIN  =  TRAP+12      (104412)
025524  025570                  .CLRDK                  ;CLRDK  =  TRAP+14      (104414)
025526  025616                  .MRDMD                  ;MRDMD  =  TRAP+16      (104416)
025530  021410                  .MRCK                   ;MRCK   =  TRAP+20      (104420)
025532  021360                  .MRCLK                  ;MRCLK  =  TRAP+22      (104422)
025534  021362                  .MRINT                  ;MRINT  =  TRAP+24      (104424)
025536  021454                  .DSCK                   ;DSCK   =  TRAP+26      (104426)
025540  021436                  .MRIND                  ;MRIND  =  TRAP+30      (104430)
025542  021502                  .XBIT                   ;XBIT   =  TRAP+32      (104432)
025544  021772                  .CLKD1                  ;CLKD1  =  TRAP+34      (104434)
025546  022046                  .CLKD0                  ;CLKD0  =  TRAP+36      (104436)
025550  022776                  .CLKR1                  ;CLKR1  =  TRAP+40      (104440)
025552  023160                  .CLKR0                  ;CLKR0  =  TRAP+42      (104442)
025554  022622                  .RBIT                   ;RBIT   =  TRAP+44      (104444)
025556  022570                  .MCLK1                  ;MCLK1  =  TRAP+46      (104446)
025560  022612                  .MCLK0                  ;MCLK0  =  TRAP+50      (104450)
025562  022370                  .MCLKB                  ;MCLKB  =  TRAP+52      (104452)
025564  025710                  .GETSP                  ;GETSP  =  TRAP+54      (104454)
025566  025750                  .SPASS                  ;SPASS  =  TRAP+56      (104456)
```

```
                                        ;CLEAR ALL DISK REGISTERS
025570  012777  000040  153304  .CLRDK: MOV     #40,@RSCS2      ;CLEAR ALL DSK REG
025576  016777  153360  153276          MOV     UNNUM,@RSCS2    ;GET UNIT NUMBER
025604  005067  153366                  CLR     MCCNT           ;CLEAR MAINT CLOCK COUNT
025610  005067  153364                  CLR     MCCNT+2
025614  000002                          RTI

025616  012777  000001  153300  .MRDMD: MOV     #1,@RSMR        ;PUT DRIVE INTO MAINT MODE
025624  000002                          RTI

025626  005067  153364          WAITRY: CLR     WORK            ;CLEAR COUNTER
025632  105777  153242          1$:     TSTB    @RSCS1          ;TEST READY
025636  100406                          BMI     2$              ;OK CONT
025640  005267  153352                  INC     WORK            ;UPDATE COUNTER
025644  005767  153346                  TST     WORK            ;DONE YET?
025650  001403                          BEQ     3$              ;READY DID NOT COME UP
025652  000767                          BR      1$              ;CONTINUE WAITING
025654  062716  000002          2$:     ADD     #2,(SP)         ;UPDATE RETURN PC
025660  000207          3$:     RTS     PC                      ;RETURN

                                        ;ROUTINE TO SHIFT COMPLETE DATA TABLE ONE BIT
                                        ;TO THE LEFT.  CARRIES BIT 15 IF ONE RVD TO BIT 0 OF THE NEXT WORD

025662  012702  026666          MDATA:  MOV     #INBUF,R2       ;GET LEFT ADDRESS OF
025666  062702  000442                  ADD     #442,R2         ;DATA TABLE
025672  012703  000220                  MOV     #220,R3         ;SETUP COUNTER FOR 200 WORDS
025676  000241                          CLC                     ;CLEAR CARRY
025700  006142          1$:     ROL     -(R2)                   ;SHIFT DATA PATTERN
025702  005303                  DEC     R3                      ;DO ALL
025704  001375                  BNE     1$                      ;WORDS
025706  000207                  RTS     PC
025710  012767  001001  153266  .GETSP: MOV     #1001,REPT      ;SETUP COUNTER
025716  104430                  MRIND                           ;SEND INDEX PULSE TO MR REG
025720  104422          1$:     MRCLK                           ;CLOCK MR
025722  005367  153256                  DEC     REPT            ;TO REACH
025726  001374                          BNE     1$              ;SECTOR PULSE
025730  032777  000400  153166          BIT     #400,@RSMR      ;DID SECTOR PULSE SET?????
025736  001401                          BEQ     2$              ;YES
025740  000002                          RTI                     ;NO REPORT ERROR
025742  062716  000002          2$:     ADD     #2,(SP)         ;UPDATE RETURN ADDR
025746  000002                          RTI

025750  104422          .SPASS: MRCLK                           ;CLOCK PAST SECTOR PULSE
025752  104422                  MRCLK
025754  005067  153216                  CLR     MCCNT           ;RESET MAINT CLOCK COUNTERS
025760  005067  153214                  CLR     MCCNT+2
025764  000002                          RTI
```

```
5014                                        ;ERROR TYPTXTOUT ROUTINE
5015
5016  025766 005767 176660          RSREG:  TST     .HLTCT          ;SHOULD WE TYPTXT GOOD AND BAD
5017  025772 001022                         BNE     8$              ;NO
5018  025774 104402 026000                  TYPE    .+2             ;.ASCIZ " BAD="
5019  026006 010046                         MOV     @#BAD,-(6)      ;PUT BAD ON STACK
5020  026010 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5021  026012 104402 026016                  TYPE    .+2             ;.ASCIZ " GOOD="
5022  026026 010146                         MOV     @#GOOD,-(6)     ;PUT GOOD ON STACK
5023  026030 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5024  026032 000402                         BR      8$              ;TYPEOUT REGISTERS
5025  026034 000167 000432                  JMP     PTDONE          ;GET OUT
5026  026040                        8$:
5027  026040 104402 026044                  TYPE    .+2             ;.ASCIZ " CS1="
5028  026052 017746 153022                  MOV     @#RSCS1,-(6)    ;PUT @#RSCS1 ON STACK
5029  026056 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5030  026060                        1$:
5031  026060 104402 026064                  TYPE    .+2             ;.ASCIZ " ER="
5032  026072 017746 153016                  MOV     @#RSER,-(6)     ;PUT @#RSER ON STACK
5033  026076 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5034  026100                        2$:
5035  026100 104402 026104                  TYPE    .+2             ;.ASCIZ " CS2="
5036  026112 017746 152764                  MOV     @#RSCS2,-(6)    ;PUT @#RSCS2 ON STACK
5037  026116 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5038  026120 032767 000200 176524           BIT     #200,.HLTCT     ;TYPTXT SECOND SET ?
5039  026126 001076                         BNE     SEEC            ;YES
5040  026130 032767 000100 176514           BIT     #AS,.HLTCT      ;TYPTXT ER ?
5041  026136 001410                         BEQ     3$              ;NO
5042  026140 104402 026144                  TYPE    .+2             ;.ASCIZ " AS="
5043  026152 017746 152740                  MOV     @#RSAS,-(6)     ;PUT @#RSAS ON STACK
5044  026156 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5045  026160 032767 000020 176464   3$:     BIT     #BA,.HLTCT      ;TYPTXT BUS ASSRESS
5046  026166 001410                         BEQ     4$              ;NO
5047  026170 104402 026174                  TYPE    .+2             ;.ASCIZ " BA="
5048  026202 017746 152700                  MOV     @#RSBA,-(6)     ;PUT @#RSBA ON STACK
5049  026206 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5050  026210 032767 000004 176434   4$:     BIT     #DA,.HLTCT      ;TYPTXT DA ?
5051  026216 001410                         BEQ     5$              ;NO
5052  026220 104402 026224                  TYPE    .+2             ;.ASCIZ " DA="
5053  026232 017746 152652                  MOV     @#RSDA,-(6)     ;PUT @#RSDA ON STACK
5054  026236 104404                         TYPEO                   ;TYPE STACK IN OCTAL
5055  026240 032767 000010 176404   5$:     BIT     #WC,.HLTCT      ;TYPTXT WC?
5056  026246 001410                         BEQ     6$              ;NO
5057  026250 104402 026254                  TYPE    .+2             ;.ASCIZ " WC="
5058  026262 017746 152616                  MOV     @#RSWC,-(6)     ;PUT @#RSWC ON STACK
5059  026266 104404                         TYPEO                   ;TYPE STACK IN OCTAL
```

```
5060  026270  032767  000040  176354  6$:     BIT     #DS,.HLTCT      ;DRIVE STATUS
5061  026276  001475                          BEQ     PTDONE          ;NO
5062  026300  104402  026304                  TYPE    .+2             ;.ASCIZ " DS="
5063  026312  017746  152574                  MOV     @RSDS,-(6)      ;PUT @RSDS ON STACK
5064  026316  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5065  026320  000167  000146                  JMP     PTDONE          ;GET OUT
5066  026324  042767  000200  176320  SEEC:   BIC     #200,.HLTCT     ;CLEAR COMMON BIT
5067  026332  032767  000240  176312          BIT     #DT,.HLTCT      ;TYPTXT DRIVE TYPE?
5068  026340  001410                          BEQ     9$              ;NO
5069  026342  104402  026346                  TYPE    .+2             ;.ASCIZ " DT="
5070  026354  017746  152546                  MOV     @RSDT,-(6)      ;PUT @RSDT ON STACK
5071  026360  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5072  026362  032767  000210  176262  9$:     BIT     #DB,.HLTCT      ;TYPTXT DATA BUFFER
5073  026370  001410                          BEQ     10$             ;NO
5074  026372  104402  026376                  TYPE    .+2             ;.ASCIZ " DB="
5075  026404  017746  152512                  MOV     @RSDB,-(6)      ;PUT @RSDB ON STACK
5076  026410  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5077  026412  032767  000220  176232  10$:    BIT     #MR,.HLTCT      ;TYPTXT MN?
5078  026420  001410                          BEQ     11$             ;NO
5079  026422  104402  026426                  TYPE    .+2             ;.ASCIZ " MR="
5080  026434  017746  152464                  MOV     @RSMR,-(6)      ;PUT @RSMR ON STACK
5081  026440  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5082  026442  032767  000204  176202  11$:    BIT     #LA,.HLTCT      ;TYPTXT LA?
5083  026450  001410                          BEQ     PTDONE          ;NO
5084  026452  104402  026456                  TYPE    .+2             ;.ASCIZ " LA="
5085  026464  017746  152430                  MOV     @RSLA,-(6)      ;PUT @RSLA ON STACK
5086  026470  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5087  026472  052767  100000  152470  PTDONE: BIS     #BIT15,ONCEE    ;SET FORND ERROR FLAG
5088  026500  032767  000040  152462          BIT     #BIT5,ONCEE
5089  026506  001466                          BEQ     1$
5090  026510  104402  026514                  TYPE    .+2             ;.ASCIZ <15><12>"MAINT CLOCK COUNT "
5091  026542  016767  152430  152460          MOV     MCCNT,WORK4     ;GET MAINT CLOCK COUNT
5092  026550  016767  152424  152446          MOV     MCCNT+2,WORK2   ;CAL NUMBERS FOR DOUBLE PRECISION
5093  026556  006167  152442                  ROL     WORK2
5094  026562  006167  152442                  ROL     WORK4
5095  026566  000241                          CLC
5096  026570  016746  152434                  MOV     WORK4,-(6)      ;PUT WORK4 ON STACK
5097  026574  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
5098  026576  012767  000005  152426          MOV     #5,WORK5
5099  026604  005067  152424          2$:     CLR     WORK6
5100  026610  006167  152410                  ROL     WORK2
5101  026614  006167  152414                  ROL     WORK6
5102  026620  006167  152400                  ROL     WORK2
5103  026624  006167  152404                  ROL     WORK6
5104  026630  006167  152370                  ROL     WORK2
5105  026634  006167  152374                  ROL     WORK6
5106  026640  016746  152370                  MOV     WORK6,-(6)      ;PUT WORK6 ON STACK
5107  026644  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
5108  026646  005367  152360                  DEC     WORK5
5109  026652  001354                          BNE     2$
5110  026654  104402  026660                  TYPE    .+2             ;.ASCIZ <15><12>
5111  026664  000207                  1$:     RTS     PC
5112  026666  000300                  INBUF:  .BLKW   300
5113  027466  000300                  OUTBUF: .BLKW   300
```

```
                                        ;THIS ROUTINE IS FOR PROGRAMMERS ONLY !!!!!!!!!!!!!THIS ROUTINE IS USED TO "DETERMINE" A
                                        ;SO THAT A 1 CAN BE ROTATED THROUGH THE CRC REGISTER BY ROTATING THE DATA PATTERN
030266  012767  000040  150650  CRCAL:  MOV     #40,FLAG2
030274  012706  000500                  MOV     #500,SP
030300  005067  150722                  CLR     WORK3
030304  012702  026666                  MOV     #INBUF,R2
030310  012701  000221                  MOV     #145.,R1
030314  005022                  1$:     CLR     (R2)+           ;CLEAR DATA BUFFER
030316  005301                          DEC     R1
030320  001375                          BNE     1$
030322  012767  000401  150676          MOV     #401,WORK3      ;START WITH A NUMBER OF 401
030330  012702  026666                  MOV     #INBUF,R2
030334  062702  000100                  ADD     #100,R2
030340  062767  000003  150660          ADD     #3,WORK3
030346  016712  150654                  MOV     WORK3,(R2)      ;PUT NUMBER INTO BUFFER
030352  012701  001001          2$:     MOV     #513.,R1        ;513=32 WORDS X 16 BITS
030356  005301                  6$:     DEC     R1
030360  001763                          BEQ     3$
030362  012700  000040                  MOV     #40,R0
030366  012702  026666                  MOV     #INBUF,R2
030372  062702  000102                  ADD     #102,R2
030376  000241                          CLC
030400  006142                  5$:     ROL     -(R2)
030402  005300                          DEC     R0
030404  001375                          BNE     5$
030406  004767  173130                  JSR     PC,GENCRC
030412  022767  000001  150576          CMP     #1,WORK
030420  001013                          BNE     4$
030422  104402  030426                  TYPE    .+2             ;.ASCIZ <15><12>"CRC= "
030436  016746  150554                  MOV     WORK,-(6)       ;PUT WORK ON STACK
030442  104404                          TYPEO                   ;TYPE STACK IN OCTAL
030444  004767  000040                  JSR     PC,TABTYP
030450  022767  000002  150540  4$:     CMP     #2,WORK
030456  001337                          BNE     6$
030460  104402  030464                  TYPE    .+2             ;.ASCIZ <15><12>"CRC= "
030474  016746  150516                  MOV     WORK,-(6)       ;PUT WORK ON STACK
030500  104404                          TYPEO                   ;TYPE STACK IN OCTAL
030502  004767  000002                  JSR     PC,TABTYP
030506  000723                          BR      6$
030510  012702  026666          TABTYP: MOV     #INBUF,R2
030514  012705  000220                  MOV     #220,R5
030520  012767  000004  150456  2$:     MOV     #4,REPT
030526                          1$:
030526  012246                          MOV     (R2)+,-(6)      ;PUT (R2)+ ON STACK
030530  104404                          TYPEO                   ;TYPE STACK IN OCTAL
030532  104402  000040                  TYPE    ,40
030536  005305                          DEC     R5
030540  001410                          BEQ     3$
030542  005367  150436                  DEC     REPT
030546  001367                          BNE     1$
030550  104402  030554                  TYPE    ,.+2            ;.ASCIZ <15><12>
030560  000757                          BR      2$
030562  000207                  3$:     RTS     PC
000001                                  .END
```

```
AOECK   020170          4048#
AS    = 000100           831#  1092  2054  2058  2061  2161  2165  2168  2424  3530  5040
ATA   = 100000           852#
BA    = 000020           829#  1062  1121  1127  1209  1213  1217  1239  1425  1429  5045
BAD   =X000C00           779#   962*  1186* 1225* 1226  1263* 1264  1300* 1301  1402* 1403  1442* 1443
                        1443*  1450  1454* 1455  1640* 1642  1652* 1654  1717* 1719  1728* 1730  1740*
                        1742   1752* 1754  1765* 1786  1806* 1808  1817* 1819  1824* 1826  1862* 1864
                        1875*  1877  1884* 1886  1892* 1894  1974* 1975  1986* 1988  1993* 1995  2000*
                        2002   2196* 2198  2247* 2249  2301* 2303  2349* 2351  2410* 2413  2436* 2896*
                        2902   2908  2910  2914  3537* 3540  3557* 3566* 3569  3652* 3654  4224* 4227
                        4236*  4239  4316* 4317  4334* 4335  4493* 4494  4518* 4519  5019
BAI   = 000010           855#  4139
BEGIN   001236           740    919#
BEGIN1  000232           736    742#
BEGIN2  000226           740#   743
BELL  = 000007           753#  4166  4753
BITBA   004442          1421#
BITCS2  004326          1397#
BITST   004254          1384#
BITWC   004372          1409#
BITO  = 000001           762#  1008  1026  3528  4156  4613  4616  4624  4637
BIT1  = 000002           763#  1848  1870  1874  1879  1881  1889  1891  1896  1899  2688  4622
BIT10 = 002000           772#  1768  1784  1801  4271  4292  4294  4549  4557  4570  4573
BIT11 = 004000           773#  4086  4442  4449  4454  4456  4487  4512  4544
BIT12 = 010000           774#  2398  2441  2444  2904  4311  4328  4330
BIT13 = 020000           775#   980   991  1012  4562  4567
BIT14 = 040000           776#  4635
BIT15 = 100000           777#   983   993  1000  3612  4640  4643  5087
BIT2  = 000004           764#  1926  1943  3593  3601  3605  3608  3611  3614  3617  4187  4189  4475
                        4502   4528  4627  4630
BIT3  = 000010           765#  1931  1947  1980  2007  2009  2011  4099  4114  4499
BIT4  = 000020           766#   928   939  1002  3130  3161  3327  3356  3692  4180
BIT5  = 000040           767#  1847  2025  2074  2086  2131  2179  2284  2509  2702  2763  2928  2978
                        3123   3180  3320  3374  3586  3687  3781  4446  4581  4594  4602  5088
BIT6  = 000100            739   742   768#  931   2596  3868  4250  4254  4477  4504
BIT7  = 000200           769#  2607  2614  2619  3879  3886  3892  4246  4252  4535
BIT8  = 000400           770#  2609  2618  2824  2826  3039  3041  3240  3242  3434  3436  3881  3891
                        4256   4260  4263  4444  4479  4488  4506  4513  4537
BIT9  = 001000           771#  1631  1633  1636  1775  1777  1779  4262  4267  4278  4295
CALRTB  023314          4468   4525#
CHCKDV  002152          1005   1008#
CHG     021234          1991   1998  2005  4194#
CLKCNT  001210           877#  4255* 4443* 4534* 4539  4542  4545  4547  4553  4555  4559  4568  4571
CLKDO = 104436          2602   3874  4958#
CLKD1 = 104434          2598   3870  4957#
CLKRO = 104442          2819   3034  3235  3429  4960#
CLKR1 = 104440          2816   3031  3232  3426  4959#
CLRDK = 104414          1151   1384  1430  1436  1464  1493  1671  1846  1883  1925  1965  2024  2073
                        2178   2214  2230  2265  2283  2316  2333  2363  2384  2460  2701  2927  3122
                        3319   3517  3565  3618  3636  3667  3686  3709  3731  3984  4060  4090  4112
                        4134   4949#
CRCAL   030266          5116#
CRCTAB  022230          4343   4355#
CRCTYP  022120          3499   4343#
CRCWD   023504          4538   4567#
CS1   = 000001           825#  1057  1116  1156  1160  1164  1172  1389  1393  1472  1477
```

```
CS2     = 000200   832#  1048  1112  1182  1192  1196  1201  1447
DA      = 000004   827#  1068  1284  1288  1292  1313  2213  2264  2419  2435  2681  2687  2884
                  2890  3100  3106  3297  3491  3602  3609  3615  3702  3706  3712  3716  3963
                  3966  3972  5050
DAO     = 001000   853#
DB      = 000210   834#  5072
DCK     = 100000   854#
DISPLA= 177570   752#  4732*  4736*
DLT     = 100000   847#
DONE      021006  1021  4160#
DONEE     002326  1003  1009  1021#
DRY     = 000200   848#
DS      = 000040   830#  1052  1872  1938  1942  1953  1957  2054  2058  2061  2161  2165  2168
                  2208  2217  2221  2259  2268  2272  2313  2319  2323  2360  2366  2370  2428
                  2884  3100  3297  3491  3497  3530  3548  3602  3609  3615  3621  3661  3665
                  3671  3675  3697  3702  3712  3716  3955  3969  4055  4058  4063  4066  4110
                  4125  4148  4152  5060
DSCK    = 104426  2036  2042  2049  2111  2117  2156  2186  2203  2237  2254  2291  2308  2340
                  2355  4019  4954#
DT      = 000240   836#  5067
DVNUM     001732   974   977#
ER      = 000002   826#  1074  1133  1321  1325  1332  1338  1345  3548
ERR     = 040000   851#
ERRORS    001002   784#  4754*  4774
E1910     022552  4383  4418#
E1915     022561  4385  4420#
E192      022536  4379  4416#
E197      022544  4381  4417#
E2410     022511  4373  4410#
E2412     022520  4375  4412#
E2415     022527  4377  4414#
E242      022467  4367  4404#
E245      022475  4369  4406#
E247      022503  4371  4408#
E3010     022442  4361  4398#
E3012     022451  4363  4400#
E3015     022460  4365  4402#
E302      022420  4355  4395#
E305      022426  4357  4396#
E307      022434  4359  4397#
FILCHR    001014   788#  4681  4682
FLAG2     001144   859#  929*  2459*  2688  2710*  2936*  3121*  3130  3161*  3318*  3327  3356*  3730*
                  4089*  4246  4252*  4260  4262*  4263*  4267  4271  4275*  4278  4292  4294*  4295*
                  4442*  4446  4449*  4454  4456*  4475  4502  4581  4594  4602  5116*
FLAG3     001146   739*  742*  860#  928*  931  939*  1002  4180
FLOTBA    003364  1222#
FLOTDA    003664  1297#
FLOTWC    003524  1260#
GENCRC    023542  2620  2828  3043  3898  4580#  5139
GETSP   = 104454  2495  2748  2964  3165  3360  3767  4965#
GOOD    =%000001   778#  963*  1187*  1222*  1224  1226  1229*  1260*  1262  1264  1267*  1297*  1299
                  1301  1304*  1400*  1401*  1403  1440*  1441*  1443  1448*  1450  1453*  1455  1641*
                  1642  1653*  1654  1718*  1719  1729*  1730  1741*  1742  1753*  1754  1785*  1786
                  1807*  1808  1818*  1819  1825*  1826  1863*  1864  1876*  1877  1885*  1886  1893*
                  1894  1973*  1975  1987*  1988  1994*  1995  2001*  2002  2197*  2198  2248*  2249
                  2302*  2303  2350*  2351  2395*  2396  2400*  2405  2406  2411*  2413  2437*  2897*
```

```
                2902    2908    2914    3538*   3539*   3540    3558*   3567*   3568*   3569    3653*   3654    4225*
                4227    4237*   4239    4313*   4314*   4317    4331*   4332*   4335    4474*   4477*   4478*   4483*
                4487*   4492*   4494    4501*   4504*   4505*   4508*   4512*   4517*   4519    4544*   4549*   4557*
                4562*   4567*   4570*   4573*   5022
HLT    = 104000  748#    964     1048    1052    1057    1062    1068    1074    1080    1086    1092    1112    1116
                1121    1127    1133    1138    1143    1156    1160    1164    1172    1182    1188    1192    1196
                1201    1209    1213    1217    1228    1239    1247    1251    1255    1266    1276    1284    1288
                1292    1303    1313    1321    1325    1332    1338    1345    1355    1362    1370    1378    1389
                1393    1405    1413    1417    1425    1429    1447    1452    1457    1472    1477    1504    1510
                1522    1526    1535    1539    1542    1551    1555    1564    1568    1578    1582    1588    1595
                1599    1610    1624    1628    1644    1656    1685    1693    1697    1700    1709    1713    1721
                1732    1744    1756    1788    1810    1821    1828    1831    1871    1872    1880    1890    1898
                1938    1942    1953    1957    1978    1992    1999    2006    2038    2044    2051    2054    2058
                2061    2083    2090    2094    2101    2105    2113    2119    2126    2135    2139    2148    2152
                2158    2161    2165    2168    2188    2202    2205    2208    2213    2217    2221    2239    2253
                2256    2259    2264    2268    2272    2293    2307    2310    2313    2319    2323    2342    2354
                2357    2360    2366    2370    2415    2419    2424    2428    2435    2438    2498    2504    2513
                2517    2563    2530    2538    2542    2550    2554    2558    2564    2568    2577    2584    2588
                2601    2603    2637    2644    2648    2656    2660    2664    2671    2675    2681    2684    2687
                2751    2758    2767    2771    2780    2784    2791    2795    2805    2817    2820    2841    2845
                2849    2853    2863    2867    2877    2884    2887    2890    2909    2967    2973    2982    2986
                2995    2999    3007    3011    3020    3032    3035    3056    3060    3064    3068    3078    3082
                3092    3100    3103    3106    3168    3175    3184    3188    3197    3201    3208    3212    3221
                3233    3236    3253    3257    3261    3265    3275    3279    3289    3297    3300    3363    3369
                3378    3382    3391    3395    3402    3406    3415    3427    3430    3447    3451    3455    3459
                3469    3473    3483    3491    3494    3497    3530    3542    3548    3552    3559    3571    3602
                3609    3615    3621    3657    3661    3665    3671    3675    3697    3702    3706    3712    3716
                3770    3776    3785    3789    3798    3802    3810    3814    3822    3826    3830    3836    3840
                3849    3853    3860    3873    3878    3913    3921    3925    3934    3938    3945    3949    3953
                3955    3963    3966    3969    3972    4021    4029    4036    4040    4052    4055    4058    4063
                4066    4107    4110    4122    4125    4148    4152
HLTADR  001012  787#    4758*   4759*   4760    4761    4775
IADONE  016312  3596    3623#
ICNT    001000  783#    926*    4179*   4710    4725    4727    4729*   4730*   4732    4735*   4736    4742    4770*
IE     = 000100  856#
ILFDON  006744  1916    1918#
INBIT   001212  878#    4606*   4608*   4610*   4614    4617    4625*   4628    4638*   4641    4657    4658*   4659*
INBUF   026666  2474    2492    2589    2622    2718    2809    2830    2896    2944    2961    3024    3045    3143
                3225    3243    3338    3419    3437    3746    3764    3861    3900    4098*   4100    4115    4140
                4141*   4584    4990    5112#   5119    5125    5133    5153
INFTST  020774  4088    4156#
INPUT   025450  937     4875    4908    4909    4929#
INT     004622  1464#
INTDON  004712  1473    1478#
INTMR   017674  3762    3961#
INTMR1  017744  3956    3973#
IR     = 000100  842#
LA     = 000204  833#    1510    1542    1610    1700    5082
LAD     001010  786#    927*    3588*   4178*   4731*   4737    4739    4742
LBT    = 002000  850#
LSTEV   001150  861#
LSTOO   001152  862#    4248*   4253*   4266*   4308
LSTWD   023436  4536    4553#
MCCNT   001176  873#    1674*   4026*   4220*   4221*   4392*   4393*   4426*   4427*   4470*   4471*   4970*   4971*
                5011*   5012*   5091    5092
MCLKB  = 104452  1618    1649    4964#
```

```
MCLKO = 104450          1523  1536  1552  1565  1579  1592  1621  1694  1710  2091  2102  2136  2149
                        2514  2527  2539  2551  2561  2574  2585  2641  2653  2661  2672  2768  2781
                        2792  2842  2850  2864  2881  2983  2996  3008  3057  3065  3079  3097  3185
                        3198  3209  3254  3262  3276  3294  3379  3392  3403  3448  3456  3470  3488
                        3786  3799  3811  3823  3933  3846  3857  3918  3931  3942  3950  4037  4325
                        4963#
MCLK1 = 104446          1519  1532  1548  1561  1575  1585  1596  1625  1690  1706  2048  2087  2098
                        2132  2145  2155  2510  2523  2535  2547  2555  2565  2581  2633  2645  2657
                        2668  2764  2777  2788  2838  2846  2860  2874  2979  2992  3004  3053  3061
                        3075  3089  3181  3194  3205  3250  3258  3272  3286  3375  3388  3399  3444
                        3452  3466  3480  3782  3795  3807  3819  3827  3837  3850  3910  3922  3935
                        3946  4033  4303  4962#
MDATA   025662          3303  3503  4990#
MPRO  = 172100           871#
MR    = 000220          835#  1080  1138  1355  1362  1370  1378  1510  1542  1610  2498  2751  2967
                        3168  3363  3770  5077
MRAOE   017750          3984#
MRCAL   023336          4315  4333  4534#
MRCILF  006442          1883#
MRCK  = 104420          1496  1502  1520  1524  1533  1537  1549  1553  1562  1566  1576  1580  1586
                        1593  1597  1622  1626  1677  1683  1691  1695  1707  1711  1829  1854  1928
                        1967  2027  2076  2081  2080  2092  2099  2103  2124  2133  2137  2146  2150
                        2181  2232  2286  2335  2386  2463  2502  2511  2515  2524  2528  2536  2540
                        2548  2552  2556  2562  2566  2575  2582  2506  2635  2642  2646  2654  2658
                        2662  2669  2673  2705  2756  2765  2769  2778  2782  2789  2793  2803  2839
                        2843  2847  2851  2861  2865  2875  2931  2971  2980  2984  2993  2997  3005
                        3009  3018  3054  3058  3062  3066  3076  3080  3090  3126  3173  3182  3186
                        3195  3199  3206  3210  3219  3251  3255  3259  3263  3273  3277  3287  3323
                        3367  3376  3380  3389  3393  3400  3404  3413  3445  3449  3453  3457  3467
                        3471  3481  3519  3590  3641  3689  3736  3774  3783  3787  3796  3800  3808
                        3812  3820  3824  3828  3834  3838  3847  3851  3858  3911  3919  3923  3932
                        3936  3943  3947  3951  3987  4027  4034  4038  4050  4951#
MRCLK = 104422          1727  1738  1751  1761  1805  1816  2041  2107  2116  4018  4048  4049  4214
                        4215  4952#  5000  5009  5010
MRCRC   014160          3122#  3304
MRCRC1  014364          3181#  3190
MRCRC2  014440          3205#  3214
MRCRC3  014510          3224#
MRCRC4  014670          3270#
MRCRC5  014734          3294#
MRDCK   015006          3319#  3504
MRDCK1  015212          3375#  3384
MRDCK2  015266          3399#  3408
MRDCK3  015336          3418#
MRDCK4  015516          3464#
MRDCK5  015562          3488#
MRDMD = 104416          1853  1927  1966  2026  2075  2180  2231  2285  2334  2385  3518  3589  3640
                        3688  4950#
MRDSEL  011146          2384#
MRD3    012726          2808#
MRD4    013126          2859#
MRD5    013172          2881#
MRD6    013230          2896#
MREX    016756          3731#
MREX1   017144          3782#  3791
MREX2   017220          3807#  3816
```

```
MREX3   017304          3833#   3842
MRIFT   015676          3517#
MRILF   006224          1846#
MRIND = 104430          1495    1500    1676    1681    1822    1860    1970    2030    2080    2123    2462    2501    2704
                        2755    2930    2970    3125    3172    3322    3366    3647    3648    3649    3735    3773    3986
                        4003    4025    4955#   4999
MRINT = 104424          1498    1679    1856    1930    1969    2029    2078    2183    2234    2288    2337    2388    2465
                        2707    2933    3128    3325    3521    3592    3643    3691    3738    3989    4953#
MRLF1   006250          1849    1853#   1917
MRLF2   006262          1861#
MRLF3   006662          1902#   1904    1906    1908    1910    1912    1914
MROP    006746          1925#
MROPER  007136          1965#   2010
MROPI   016314          3636#
MROPIA  016506          3664    3667#
MRPAR   016630          3686#
MRRD    012412          2701#
MRRD1   012602          2764#   2773
MRRD2   012656          2788#   2797
MRRT1   007726          2087#   2096
MRSRC   007662          2073#
MRSRCH  007500          2024#
MRTIME  004714          1493#
MRTIM1  004762          1519#   1528
MRT2    005040          1541    1546#
MRT2A   005050          1548#   1557
MRT2B   005104          1561#   1570
MRT2C   005140          1575#   1584
MRT3    005256          1617#   1647
MRT4    005450          1671#
MRT4A   005522          1690#   1702
MRT4B   005572          1710#   1798
MRT4C   005624          1727#   1734
MRT4D   005676          1750#   1790
MRT4E   005704          1751#   1758    1802
MRT4F   006124          1763    1804#
MRT4G   006156          1816#
MRVR    020254          4086#
MRVRR   020560          4124    4131#
MRVRI   020410          4109    4112#
MRVR2   020564          2690    4134#
MRWCK   013346          2927#
MRWCK1  013522          2979#   2988
MRWCK2  013576          3004#   3013
MRWCK3  013646          3023#
MRWCK4  014046          3074#
MRWCK5  014112          3097#
MRWRT   011466          2460#   4131
MRWRT1  011626          2510#   2519
MRWRT2  011702          2535#   2544
MRWRT3  011766          2561#   2570
MRWRI   010076          2132#   2141
MULSYS  021066          4170    4176#   4856
MULTII  001702           933     938     970#   4182
N     = 000065           747#   1029    1033#   1093    1097#   1144    1148#   1165    1166#   1174    1178#   1203    1204#
                        1219    1220#   1231    1232#   1240    1241#   1256    1257#   1270    1271#   1277    1278#   1293
```

DERSCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    1294#  1307   1308#  1314   1315#  1327   1328#  1333   1334#  1339   1340#  1346   1347#
                    1356   1357#  1363   1364#  1371   1372#  1379   1383#  1395   1396#  1407   1408#  1419
                    1420#  1431   1435#  1460   1464#  1479   1483#  1657   1661#  1832   1836#  1919   1923#
                    1959   1963#  2012   2016#  2062   2066#  2169   2173#  2222   2226#  2274   2278#  2324
                    2328#  2371   2375#  2446   2450#  2691   2695#  2917   2921#  3107   3111#  3306   3310#
                    3506   3510#  3574   3578#  3624   3628#  3676   3680#  3718   3722#  3974   3978#  4079
                    4083#
NED     = 010000     845#  2426
NEDDON  011366      2402   2441#
NMDD    011456      2434   2439   2442   2445#
NOPERR  021140      1937   1941   1952   1956   4187#
NOWEV   001154       863#
NOWGO   002332       967   1001   1020   1026#  4183
NOWOD   001156       864#  4258*  4266   4277*  4282*  4289*  4291*  4305   4326   4451*  4458*  4460*  4464*
                    4466*  4526
ONCEE   001170       869#   930*   983*   993   1000*  1008   1026*  1494*  1631   1633*  1636*  1672*  1673*
                    1768   1775   1777*  1779*  1784*  1801*  1847*  1848   1870*  1874*  1879*  1881*  1889*
                    1891*  1896*  1899*  1926*  1931   1943*  1947*  1958*  1980   2007   2009*  2011*  2025*
                    2074*  2086*  2131*  2179*  2284*  2441   2444*  2461*  2509*  2596*  2607   2609   2614*
                    2618*  2619*  2702*  2703*  2763*  2824   2826*  2928*  2929*  2978*  3039   3041*  3123*
                    3124*  3180*  3240   3242*  3320*  3321*  3374*  3434   3436*  3586*  3593   3601*  3605*
                    3608*  3611*  3614*  3617*  3687*  3734*  3781*  3868*  3879   3881   3886*  3891*  3892*
                    4156*  4187   4189*  4250   4254*  4256   4444   4479   4488   4506   4513   4535   4537
                    5087*  5088
OR      = 000200     843#
OUTBUF  027466      2740   2744   2897   3156   3160   3351   3355   3525   3555   3558   3637   3993   3996
                    5113#
PC      =%000007     761#  1937*  1941*  1952*  1956*  1991*  1998*  2005*  2620*  2828*  3043*  3303*  3499*
                    3503*  3898*  4193*  4197*  4315*  4333*  4350*  4352*  4356*  4358*  4360*  4362*  4364*
                    4366*  4368*  4370*  4372*  4374*  4376*  4378*  4380*  4382*  4384*  4386*  4468*  4529*
                    4550*  4563*  4574*  4612*  4626*  4639*  4652*  4660*  4764*  4941*  4985*  4997*  5111*
                    5139*  5145*  5151*  5166*
PCNT    001004       785#  4161*  4162*  4172
PGE     = 002000     844#
PGTRAP  004676      1465   1474#
PIP     = 020000     849#
PS      = 177776     749#   750   1033*  1098*  1467*  3732*
PSW     = 177776     750#
PTDONE  026472      5025   5061   5065   5083   5087#
Q0      = 000001    1029#  1093#  1144#  1165#  1174#  1203#  1219#  1231#  1240#  1256#  1270#  1277#  1293#
                    1307#  1314#  1327#  1333#  1339#  1346#  1356#  1363#  1371#  1379#  1395#  1407#  1419#
                    1431#  1460#  1479#  1657#  1832#  1919#  1959#  2012#  2062#  2169#  2222#  2274#  2324#
                    2371#  2446#  2691#  2917#  3107#  3306#  3506#  3574#  3624#  3676#  3718#  3974#  4079#
RBIT    = 104444     2815   3030   3231   3425   4961#
RDLIN   = 104412      936   4873   4948#
RDOCT   = 104410      942   4947#
REGCHG  021250      4194   4199#
REPT    001204       875#  1518*  1527*  1547*  1556*  1560*  1569*  1574*  1583*  1617*  1619*  1648*  1650*
                    1689*  1701*  1726*  1733*  1749*  1789*  1800*  1804*  1811*  2040*  2045*  2085*  2095*
                    2106*  2108*  2115*  2120*  2130*  2140*  2508*  2518*  2534*  2543*  2560*  2569*  2640*
                    2649*  2762*  2772*  2787*  2796*  2812*  2822*  2836*  2859*  2868*  2898*  2906*  2977*
                    2987*  3003*  3012*  3027*  3037*  3051*  3074*  3083*  3179*  3189*  3204*  3213*  3228*
                    3238*  3248*  3270*  3280*  3373*  3383*  3398*  3407*  3422*  3432*  3442*  3464*  3474*
                    3780*  3790*  3806*  3815*  3832*  3841*  3917*  3926*  3997*  3999*  4007*  4015*  4032*
                    4041*  4101*  4105*  4116*  4120*  4481   4560*  4583*  4646*  4998*  5001*  5155*  5162*
REPT1   001206       876#  1616*  1646*  1750*  1757*  1799*  2811*  2827   2831*  3026*  3042   3046*  3227*
```

DERSCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                3421*  4484*  4486*  4490   4509*  4511*  4515   4593*  4596*  4604   4644*
RMRC1  010220   2178#
RMRC2  010404   2230#
RMRC3  010560   2283#
RMRC4  010732   2333#
RSAS   001116    803#    960    962    977   1090   1974   1993   2055*  2056   2162*  2163   2346*  2422
                3523*  5043
RSBA   001106    799#   1036*  1060   1102*  1119   1206*  1207   1210*  1211   1214*  1215   1224*  1225
                1235*  1236*  1237   1421*  1423   1426*  1427   2492*  2740*  2961*  3156*  3351*  3525*
                3555   3557   3637*  3764*  3993*  4100*  4115*  4140*  5048
RSBAB  001142    813#   1422*
RSCS1  001100    796#   1035*  1055   1100*  1114   1153*  1154   1157*  1158   1161*  1162   1168*  1169*
                1170   1385*  1387   1390*  1391   1468*  1475   1861*  1933*  1949*  1982*  1985*  2035*
                2052   2059   2110*  2159   2166   2185*  2192*  2206   2219   2236*  2257   2270   2290*
                2311   2321   2339*  2358   2368   2404*  2417   2494*  2679   2742*  2882   2963*  3098
                3158*  3295   3353*  3489   3527*  3528   3546   3598*  3612   3645*  3663   3700   3710
                3766*  3961   4002*  4102*  4103   4108   4117*  4118   4123   4142*  4143   4150   4978
                5028
RSCS1B 001134    810#   1386*  2425*
RSCS2  001102    797#    956*   971*   975*   986*  1034*  1041*  1046   1049*  1099*  1101*  1107*  1109*
                1110   1113*  1180   1183*  1184   1186   1189*  1190   1193*  1194   1197*  1198*  1199
                1202*  1397*  1398*  1402   1442   2396*  2398   2406*  2410   2426   2430*  3537   3566
                3692*  4099*  4114*  4139*  4145   4968*  4969*  5036
RSCS2B 001136    811#   1399*
RSDA   001110    800#   1037*  1066   1103*  1125   1281*  1282   1285*  1286   1289*  1290   1299*  1300
                1309*  1310*  1311   2031*  2034*  2184*  2211   2235*  2243*  2262   2289*  2338*  2390*
                2407*  2431   2685   2888   3104   3597*  3693*  3704   3964   3970   3991*  5053
RSDB   001122    805#   1105*  1437*  1438*  1449   1454   5075
RSDS   001112    801#   1050   1862   1884   1939   1954   2000   3606   3659   3672   4056   4064   4236
                5063
RSDT   001126    807#    987    989   1603   1629   1766   2032   5070
RSER   001114    802#    957*   972*  1038*  1072   1104*  1131   1318*  1319   1322*  1323   1329*  1330
                1335*  1336   1341*  1342*  1343   1875   1892   1935   1950   1972*  1986   2196   2215
                2247   2266   2297*  2301   2317   2349   2364   2397   3495   3522*  3599   3619   3652
                3668   3695   3714   3897*  3967   4053   4061   5032
RSLA   001120    804#   1508   1540   1608   1640   1652   1698   1717   1728   1740   1752   1765   1806
                1817   1824   5085
RSMR   001124    806#   1039*  1078   1108*  1136   1350*  1351   1358*  1359*  1360   1365*  1366   1373*
                1374   2801*  2802*  3016*  3017*  3217*  3218*  3411*  3412*  4011*  4012*  4218*  4219*
                4224   4233*  4234*  4316   4334   4390*  4391*  4425*  4432*  4473*  4493   4500*  4518
                4974*  5003   5080
RSREG  025766    4764   5016#
RSVCPS 001132    809#   1466*  3763*
RSVEC  001130    808#   1465*  3762*
RSWC   001104    798#   1040*  1084   1106*  1141   1244*  1245   1248*  1249   1252*  1253   1262*  1263
                1272*  1273*  1274   1409*  1411   1414*  1415   2493*  2682   2741*  2885   2962*  3101
                3157*  3298   3352*  3492   3526*  3550   3638*  3765*  3992*  4097*  4113*  4138*  5058
RSWCB  001140    812#   1410*
RSD    001160    865#   4617*  4625   4638
R0     =%000000  754#   4166*  4833   4852*  5132*  5137*
R1     =%000001  755#    970*   973*  4834   4844*  4845*  4851*  4870   4874*  4885*  4887*  4889*  4891
                4894*  5120*  5122*  5129*  5130*
R2     =%000002  756#    947*   949*   954    955   1546*  1605*  1607*  1608   1634*  1637*  1639*  1641
                1675*  1718   1729   1739*  1741   1753   1762   1764*  1780*  1782*  1783*  1785   1794
                1807   1850*  1861   1867   1902*  1903   1905   1907   1909   1911   1913   1915   2474*
                2475*  2476*  2481*  2486*  2622*  2623*  2624*  2625   2718*  2719*  2720*  2725*  2729*
```

DERSCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    2743*  2746*  2830*  2832*  2833*  2834.  2944*  2945*  2946*  2951*  2955*  3045*  3047*
                    3048*  3049   3143*  3145*  3148*  3149*  3151*  3159*  3163*  3243*  3244*  3245*  3246
                    3338*  3340*  3343*  3344*  3346*  3354*  3358*  3437*  3438*  3439*  3440   3587*  3595*
                    3597   3746*  3747*  3748*  3753*  3757*  3900*  3901*  3902*  3903   4009*  4011   4835
                    4850*  4871   4876*  4878   4880   4882*  4883*  4884*  4886*  4888*  4893*  4990*  4991*
                    4994*  5119*  5121*  5125*  5126*  5128*  5133*  5134*  5136*  5153*  5157
R3      =%000003     757#   1794*  1795*  1796   2477*  2479*  2481   2483*  2487*  2721*  2723*  2725   2727*
                    2730*  2744*  2745*  2947*  2949*  2951   2953*  2956*  3144*  3146*  3150*  3152*  3160*
                    3162*  3339*  3341*  3345*  3347*  3355*  3357*  3749*  3751*  3753   3755*  3758*  4010*
                    4012   4304*  4311*  4314   4328*  4330*  4332   4469*  4472*  4473   4478   4498*  4499*
                    4500   4505   4656*  4657*  4659   4636   4849*  4872   4875*  4892*  4992*  4995*
R4      =%000004     758#    943*   944    946    950    952*  2485*  2486   2590*  2728*  2729   2954*  2955
                    3756*  3757   3862*  4008*  4013*  4274*  4281*  4290*  4441*  4459*  4465*  4585*  4607*
                    4649*  4671   4691*  4685*  4699*  4788   4791*  4811   4817*  4837   4848*
R5      =%000005     759#   2589*  2590   2625*  2626*  2809*  2810*  2834*  2835*  2899*  2900*  3024*  3025*
                    3049*  3050*  3225*  3226*  3246*  3247*  3419*  3420*  3440*  3441*  3861*  3862   3903*
                    3904*  3996*  3998*  4273*  4274   4440*  4441   4584*  4585   4648*  4649   4672   4673*
                    4674   4676*  4677   4679   4687   4695*  4696*  4697   4698*  4789   4790*  4797*  4799*
                    4801*  4816*  4838   4847*  4907   4908*  4909   4926*  5154*  5160*
SAVEE    001174     872#   2827*  2831   3042*  3046   3132*  3245   3301*  3329*  3501*  4345
SC     = 100000     841#
SCOPE  = 104400    1032   1096   1147   1165   1177   1203   1219   1231   1240   1256   1270   1277   1293
                    1307   1314   1327   1333   1339   1346   1356   1363   1371   1382   1395   1407   1419
                    1434   1463   1482   1660   1835   1922   1962   2015   2065   2172   2225   2277   2327
                    2374   2449   2694   2920   3110   3309   3509   3577   3627   3679   3721   3977   4082
                    4160   4943#
SEEC     026324    5039   5066#
SP      =%000006     760#    919*  3733*  3985*  4225   4226*  4229*  4237   4238*  4241*  4319*  4337*  4496*
                    4521*  4839   4843*  4984*  5006*  5117*
SPASS  = 104456    2499   2752   2968   3169   3364   3771   4966#
STTEST   002016     981    984#  1007
SWR    = 177570     751#    752    980    991   1012   2904   4086   4163   4708   4710   4712   4723   4751
                    4755   4765   4768
SW10   = 002000     730#   4163   4751
SW11   = 004000     729#   4723
SW12   = 010000     728#   4164   4166
SW13   = 020000     727#   4755
SW14   = 040000     726#   4712
SW15   = 100000     725#
SW8    = 000400     733#   4708
SW9    = 001000     732#   4768
TABTYP   030510    5145   5151   5153#
TBDIA    020476    4111   4129#
TIMES    024514    1027   1028*  1152*  4727   4742#
TIMSV    001172     870#   1027*  1152
TPB      001020     790#   4682*  4687*
TPS      001016     789#   4683   4688
TRE    = 040000     840#
TRMR     021305    2200   2251   2305   4205#
TRY      001712     972#    976
TRYNX    002126     985    990   1002#  4157
TSTEVB   022016    4306   4309   4313#
TST1     002354    1032#
TST10    003414    1231#
TST11    003440    1240#
TST12    003522    1256#
```

DERSCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TST13   003554          1270#
TST14   003600          1277#
TST15   003662          1293#
TST16   003714          1307#
TST17   003740          1314#
TST2    002576          1096#
TST20   004002          1327#
TST21   004024          1333#
TST22   004046          1339#
TST23   004072          1346#
TST24   004130          1356#
TST25   004156          1363#
TST26   004214          1371#
TST27   004252          1382#
TST3    003020          1147#
TST30   004324          1395#
TST31   004370          1407#
TST32   004440          1419#
TST33   004512          1434#
TST34   004620          1463#
TST35   004712          1482#
TST36   005446          1660#
TST37   006222          1835#
TST4    003112          1165#
TST40   006744          1922#
TST41   007134          1962#
TST42   007476          2015#
TST43   007660          2065#
TST44   010216          2172#
TST45   010402          2225#
TST46   010556          2277#
TST47   010730          2327#
TST5    003140          1177#
TST50   011144          2374#
TST51   011456          2449#
TST52   012410          2694#
TST53   013344          2920#
TST54   014150          3110#
TST55   014776          3309#
TST56   015674          3509#
TST57   016114          3577#
TST6    003300          1203#
TST60   016312          3627#
TST61   016626          3679#
TST62   016746          3721#
TST63   017746          3977#
TST64   020252          4082#
TST7    003362          1219#
TTAGG   002600          1098#
TYPE  = 104402           935   941   958   982   995   999  1016  1017  1866  1888  1897  1977  1990
                        1997  2004  2200  2201  2251  2252  2305  2306  2353  2443  2911  3498  3656
                        3670  4130  4147  4165  4177  4190  4194  4351  4355  4357  4359  4361  4363
                        4365  4367  4369  4371  4373  4375  4377  4379  4381  4383  4385  4753  4757
                        4763  4815  4855  4898  4918  4921  4925  4944#  5018  5021  5027  5031  5035
                        5042  5047  5052  5057  5062  5069  5074  5079  5084  5090  5110  5142  5148
                        5159  5164
```

```
TYPE0 = 104404          4762  4945#  5020  5023  5029  5033  5037  5044  5049  5054  5059  5064  5071
                        5076  5081  5086  5144  5150  5158
TYPES = 104406           998  1015  1869  2913  4192  4196  4946#  5097  5107
UNCMP   001166           868#   955*   978*   984  1004*  1010*  1018*  1973  1994
UNITSV  001164           867#   954*   960   963   977*   984  2055  2162  3523
UNNUM   001162           866#   946*   956   979*   986   997  1006*  1011*  1014  1034  1049  1099  1113
                        1202  1397  1400  1441  2430  2437  3539  3568  4969
WAITRY  025626          4977#
WC    = 000010           828#  1086  1143  1247  1251  1255  1276  1413  1417  2681  2684  2884  2887
                        3100  3103  3297  3300  3491  3494  3497  3552  3665  3955  3963  3969  4110
                        4125  5055
WCE   = 040000           846#  4145
WCRC    001202           874#  4597*  4622  4635
WK15    001214           879#  4598*  4601*  4611
WORK    001216           880#  1351*  1352*  1353  1366*  1367*  1368  1374*  1375*  1376  1439*  1445*  1469*
                        1470*  1867*  1868  1934*  1948*  1979*  1984*  2394*  2401*  2591*  2611*  2615*  2616*
                        2624  2627*  2628*  2833  2910*  2912  3048  3532*  3533*  3863*  3883*  3887*  3888*
                        3902  3905*  3906*  4135*  4136*  4191  4195  4343*  4347*  4350  4472  4498  4525*
                        4528*  4611*  4621*  4624*  4634*  4637*  4651*  4656  4658  4977*  4980*  4981  5140
                        5143  5146  5149
WORK0   001220           881#  4586*  4597  4600*  4613*  4616*  4627*  4630*  4640*  4643*  4651
WORK1   001222           882#  2405*  2436  4344*  4345  4348*
WORK2   001224           883#  5092*  5093*  5100*  5102*  5104*
WORK3   001226           884#  2808*  3023*  3224*  3418*  4264*  4269  4283*  4438  4448*  4452*  4461*  5118*
                        5124*  5127*  5128
WORK4   001230           885#  5091*  5094*  5096
WORK5   001232           886#  4091*  4092*  5098*  5108*
WORK6   001234           887#  5099*  5101*  5103*  5105*  5106
XBIT  = 104432          2597  3869  4956#
XXOR    024142          4612  4626  4639  4656#
ZERONE  004514          1436#
.     = 030564           734#   735#   738#   781#   792#   935   936#   941   958   959#   982   995  1017
                       1018#  1047  1051  1056  1061  1067  1073  1079  1085  1091  1111  1115  1120
                        1126  1132  1137  1142  1155  1159  1163  1171  1181  1191  1195  1200  1208
                        1212  1216  1227  1238  1246  1250  1254  1265  1275  1283  1287  1291  1302
                        1312  1320  1324  1331  1337  1344  1354  1361  1369  1377  1388  1392  1404
                        1412  1416  1424  1428  1451  1456  1476  1509  1655  1699  1866  1888  1889#
                        1897  1898#  1977  1978#  1990  1997  2004  2060  2167  2201  2220  2252  2253#
                        2271  2306  2307#  2322  2353  2364#  2369  2414  2418  2423  2427  2443  2683
                        2686  2689  2886  2889  2911  3102  3105  3299  3493  3498  3499#  3551  3570
                        3620  3656  3670  3674  3696  3701  3705  3711  3715  3962  3965  3968  3971
                        4065  4130  4147  4148#  4151  4165  4166#  4172  4177  4190  4212#  4351  4684
                        4689  4742  4757  4758#  4763  4764#  4766  4774  4775  4784  4804  4807  4812
                        4822#  4855  4859  4898  4912  4918  4929#  5018  5021  5022#  5027  5031  5032#
                        5035  5042  5043#  5047  5048#  5052  5053#  5057  5058#  5062  5063#  5069  5070#
                        5074  5075#  5079  5080#  5084  5085#  5090  5091#  5110  5111#  5112#  5113#  5142
                        5148  5164  5165#
.CLKDO  022046          4325#  4958
.CLKD1  021772          4303#  4957
.CLKRD  023160          4498#  4960
.CLKR1  022776          4468#  4959
.CLRDK  025570          4949  4968#
.DSCK   021454          4236#  4954
.GETSP  025710          4965  4998#
.HLT    024516           922  4751#
.HLTCT  024652          4760*  4774#  5016  5038  5040  5045  5050  5055  5060  5066*  5067  5072  5077
```

F11


```
                    5082
.ILLUP  025214      4831    4858#
.KIT    024466      4713    4722    4728    4735#   4772
.MCLKB  022370      4390#   4964
.MCLKO  022612      4432#   4963
.MCLKI  022570      4425#   4962
.MRCK   021410      4224#   4951
.MRCLK  021360      4218#   4952
.MROMD  025616      4950    4974#
.MRIND  021436      4233#   4955
.MRINT  021352      4214#   4953
.OVER   024472      4711    4736#
.POWER  025066       920    4831#   4853
.POWUP  025132      4840    4843#
.PR     025042      4781*   4783*   4785*   4786*   4791    4795    4805*   4806    4809*   4811    4815    4822#
.PRF    024750      4793    4796    4801#
.PRL    024726      4794#   4810
.PTIT   024706      4782    4787#
.PUVEC  025222      4831*   4832*   4840*   4862#
.QQ.  = 000060      4930#   4943    4944#   4945#   4946#   4947#   4948#   4949#   4950#   4951#   4952#   4953#   4954#
                    4955#   4956#   4957#   4958#   4959#   4960#   4961#   4962#   4963#   4964#   4965#   4966#   4967#
.RBIT   022622      4438#   4961
.RDLIN  025344      4907#   4948
.RDOCT  025226      4868#   4947
.SAVR6  025220      4839*   4843    4861#
.SCOPE  024324      4708#   4943
.SPASS  025750      4966    5009#
.SVLAD  024446      4719    4724    4730#   4738
.SWOPT= 163400       718#    725     726     727     728     729     730     731     732     733     734     4163    4164
                    4166    4708    4751    4755    4768
.TBIT   021064      4172#
.TRAP   025470       924    4937#
.TRP    025506      4940    4941#
.TYPE   024166      4671#   4944
.TYPEB  024654      4781#
.TYPEO  024664      4783#   4945
.TYPES  024674      4785#   4946
.XBIT   021502      4246#   4956
```

DERSCB.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
END      1#  718#
NEWTST   1#  718#  1029  1093  1144  1165  1174  1203  1219  1231  1240  1256  1270  1277  1293
        1307 1314  1327  1333  1339  1346  1356  1363  1371  1379  1395  1407  1419  1431  1460
        1479 1657  1832  1919  1959  2012  2062  2169  2222  2274  2324  2371  2446  2691  2917
        3107 3306  3506  3574  3624  3676  3718  3974  4079
ODT11    1#
ODT11X   1#
POP      1#  778#  4718  4721  4816  4847  4892
PRINT    1#  778#   935   941   958   982   995  1017  1866  1888  1897  1977  1990  1997  2004
        2201 2252  2306  2353  2443  2911  3498  3656  3670  4130  4147  4165  4177  4190  4351
        4757 4763  4855  4897  4917  5018  5021  5027  5031  5035  5042  5047  5052  5057  5062
        5069 5074  5079  5084  5090  5110  5142  5148  5164
PUSH     1#  778#  4715  4787  4833  4870
SCOPE.   1# 1029# 1093# 1144# 1165# 1174# 1203# 1219# 1231# 1240# 1256# 1270# 1277# 1293# 1307#
        1314# 1327# 1333# 1339# 1346# 1356# 1363# 1371# 1379# 1395# 1407# 1419# 1431# 1460# 1479#
        1657# 1832# 1919# 1959# 2012# 2062# 2169# 2222# 2274# 2324# 2371# 2446# 2691# 2917# 3107#
        3306# 3506# 3574# 3624# 3676# 3718# 3974# 4079#
SET     4930# 4943  4944  4945  4946  4947  4948  4949  4950  4951  4952  4953  4954  4955  4956
        4957  4958  4959  4960  4961  4962  4963  4964  4965  4966
TYPADR   1#
TYPBIT   1#  778#
TYPCHR   1#  778#
TYPDEC   1#  778#
TYPLIN   1#  778#
TYPOCS   1#  778#   996  1014  1868  2912  4191  4195  5096  5106
TYPOCT   1#  778#  4761  5019  5022  5028  5032  5036  5043  5048  5053  5058  5063  5070  5075
        5080 5085  5143  5149  5156
TYPTXT   1#  778#   934   940   958   982   995  1017  1866  1888  1897  1977  1990  1997  2004
        2201 2252  2306  2353  2443  2911  3498  3656  3670  4129  4147  4176  4190  4351  5018
        5021 5026  5030  5034  5042  5047  5052  5057  5062  5069  5074  5079  5084  5090  5110
        5142 5148  5164
$CATCH   1#  718#   734
$CMTAG   1#  718#   781
$DONE    1#  718#  4158
$EQUAT   1#  718#   747
$HLT     1#  718#  4743
$IOMMR   1#
$OCTAL   1#  718#  4775
$POWER   1#  718#  4823
$RAND2   1#
$RAND4   1#
$RDDEC   1#
$RDLIN   1#  718#  4900
$RDOCT   1#  718#  4863
$SCOPE   1#  718#  4701
$SET    4930# 4943  4944  4945  4946  4947  4948  4949  4950  4951  4952  4953  4954  4955  4956
        4957  4958  4959  4960  4961  4962  4963  4964  4965  4966
$SETUP   1#  718#   919
$SMMR    1#
$SWDOC   1#  718#   722
$TRAP    1#  718#  4930
$TYPE    1#  718#  4661
$TYPEA   1#
$TYPED   1#
$UMMR    1#
```

```
ADC     4162  4221  4393  4427  4471
ADD     1605  1607  1637  1639  1782  1902  2623  2832  3047  3244  3438  3901  4161  4220  4226
        4229  4236  4241  4273  4319  4337  4347  4392  4426  4440  4470  4496  4521  4648  4692
        4695  4940  4984  4991  5006  5126  5127  5134
ASL     1004
BCC     1230  1268  1305
BCS     1005  2402  2480  2724  2950  3302  3502  3596  3752
BEQ      932   938   951   961   974   985   988   994  1047  1051  1056  1061  1067  1073  1079
        1085  1091  1111  1115  1120  1126  1132  1137  1142  1155  1159  1163  1171  1181  1185
        1191  1195  1200  1208  1212  1216  1227  1238  1246  1250  1254  1265  1275  1283  1287
        1291  1302  1312  1320  1324  1331  1337  1344  1354  1361  1369  1377  1388  1392  1404
        1412  1416  1424  1428  1444  1451  1456  1476  1509  1541  1604  1609  1630  1632  1643
        1655  1699  1720  1731  1743  1755  1763  1767  1769  1787  1797  1809  1820  1827  1865
        1878  1887  1895  1904  1906  1908  1910  1912  1914  1916  1936  1940  1951  1955  1976
        1989  1996  2003  2033  2053  2057  2060  2160  2164  2167  2199  2207  2212  2216  2220
        2250  2258  2263  2267  2271  2304  2312  2318  2322  2352  2359  2365  2369  2414  2418
        2423  2427  2434  2683  2686  2883  2886  2889  2901  2903  2907  3099  3102  3105  3296
        3299  3490  3493  3496  3529  3541  3547  3551  3556  3570  3600  3607  3620  3655  3660
        3664  3669  3674  3696  3701  3705  3711  3715  3962  3965  3968  3971  4054  4057  4062
        4065  4146  4167  4247  4327  4447  4455  4476  4480  4482  4489  4503  4507  4514  4527
        4546  4582  4595  4615  4623  4629  4636  4642  4647  4678  4709  4711  4726  4738  4752
        4804  4807  4877  4910  4982  5004  5041  5046  5051  5056  5061  5068  5073  5078  5083
        5089  5131  5161
BGT     4728  4879
BHIS    4554
BIC      742   928   929   930   983  1000  1352  1367  1375  1633  1673  1779  1780  1784  1795
        1847  1874  1881  1891  1899  1926  1943  1958  2011  2179  2284  2619  2703  2929  3124
        3321  3586  3605  3611  3617  3687  3892  4252  4254  4262  4263  4295  4330  4449  4456
        4483  4499  4508  4549  4557  4562  4567  4570  4573  4613  4627  4640  4657  4658  4696
        5066
BICB    4914
BIS      739   939  1026  1398  1401  1441  1494  1636  1672  1777  1801  1870  1879  1889  1896
        1947  2009  2025  2074  2086  2131  2411  2444  2509  2596  2614  2618  2702  2763  2826
        2928  2978  3041  3123  3161  3180  3242  3320  3356  3374  3436  3539  3568  3601  3608
        3614  3692  3781  3868  3886  3891  4099  4114  4139  4156  4189  4275  4294  4311  4314
        4328  4332  4442  4472  4477  4478  4487  4492  4498  4504  4505  4512  4517  4528  4544
        4616  4624  4630  4637  4643  4659  5087
BISB    4808
BIT      931   980   984   991   993  1002  1008  1012  1078  1114  1131  1136  1360  1631  1768
        1775  1848  1931  1980  2007  2398  2426  2441  2607  2609  2688  2824  2904  3039  3130
        3240  3327  3434  3528  3593  3612  3879  3881  4086  4145  4163  4180  4187  4246  4250
        4256  4260  4267  4271  4278  4292  4345  4444  4446  4454  4475  4479  4488  4502  4506
        4513  4535  4537  4581  4594  4602  4622  4635  4674  4708  4712  4723  4751  4755  4768
        4795  5003  5038  5040  5045  5050  5055  5060  5067  5072  5077  5082  5088
BLOS     945  4540  4543  4605
BLT     4881
BMI     4104  4119  4979
BNE      981   990   992  1003  1009  1013  1446  1471  1528  1557  1570  1584  1620  1647  1651
        1702  1734  1758  1776  1790  1812  1849  1932  1981  2008  2046  2096  2109  2121  2141
        2399  2442  2488  2519  2544  2570  2608  2610  2612  2617  2629  2650  2689  2731  2747
        2773  2797  2823  2825  2869  2905  2957  2988  3013  3038  3040  3084  3131  3147  3153
        3164  3190  3214  3239  3241  3281  3328  3342  3348  3359  3384  3408  3433  3435  3475
        3536  3594  3613  3759  3791  3816  3842  3880  3882  3884  3889  3907  3927  4000  4014
        4016  4042  4087  4093  4106  4121  4137  4164  4181  4188  4228  4240  4251  4257  4261
        4268  4270  4272  4279  4293  4306  4309  4318  4336  4346  4439  4445  4485  4491  4495
        4510  4516  4520  4536  4538  4548  4556  4560  4569  4572  4603  4645  4675  4680  4686
```

```
        4694  4713  4724  4756  4769  4796  4810  4812  4846  4916  4923  4996  5002  5017  5039
        5109  5123  5138  5141  5147  5163
BPL     2680  4109  4124  4144  4151  4684  4689  4766  4912
BR       743   953   976  1001  1007  1020  1606  1635  1638  1778  1781  1983  2403  2439  2482
        2726  2837  2915  2952  3052  3249  3443  3622  3754  3956  4111  4149  4276  4329  4349
        4450  4457  4467  4541  4558  4561  4609  4650  4690  4714  4719  4722  4782  4784  4793
        4859  4890  4899  4919  4983  5024  5152  5165
CLC     1223  1261  1298  3500  4280  4599  4993  5095  5135
CLR      926   927   947   971  1011  1169  1198  1236  1273  1310  1342  1359  1437  1448  1546
        1674  1675  1825  1893  2243  2346  2395  2407  2475  2477  2710  2719  2721  2745  2808
        2945  2947  3023  3145  3151  3162  3224  3340  3346  3357  3418  3747  3749  4026  4178
        4179  4255  4258  4277  4289  4304  4443  4451  4458  4464  4525  4586  4598  4606  4610
        4621  4634  4785  4844  4874  4970  4971  4977  5011  5012  5099  5118  5121
CLRB    1399  4770  4792  4794  4814  4924
CMP      944   960   987   989  1046  1050  1055  1084  1110  1141  1154  1158  1162  1170  1180
        1184  1190  1194  1199  1207  1211  1215  1226  1245  1249  1253  1264  1282  1286  1290
        1301  1319  1323  1330  1336  1353  1368  1376  1387  1391  1403  1411  1415  1423  1427
        1443  1450  1455  1474  1475  1603  1608  1629  1642  1654  1698  1719  1730  1742  1754
        1762  1766  1786  1796  1808  1819  1826  1864  1877  1886  1894  1903  1905  1907  1909
        1911  1913  1915  1939  1954  1975  1988  1995  2002  2032  2052  2059  2159  2166  2198
        2206  2211  2219  2249  2257  2262  2270  2303  2311  2321  2351  2358  2368  2413  2417
        2431  2685  2882  2888  2902  2908  2914  3098  3104  3295  3489  3495  3540  3546  3550
        3555  3569  3599  3606  3654  3659  3663  3672  3695  3700  3704  3710  3961  3964  3967
        3970  4053  4056  4064  4227  4239  4317  4335  4481  4490  4494  4515  4519  4539  4542
        4545  4547  4553  4555  4559  4568  4571  4604  4693  4720  4811  4909  5140  5146
CMPB     937  4679  4710  4727  4878  4880  4915  4922
DEC      952   973  1445  1470  1527  1556  1569  1583  1619  1646  1650  1701  1733  1757  1789
        1811  2045  2095  2108  2120  2140  2487  2518  2543  2569  2611  2616  2628  2649  2730
        2746  2772  2796  2822  2868  2900  2906  2956  2987  3012  3037  3083  3146  3152  3163
        3189  3213  3238  3280  3341  3347  3358  3383  3407  3432  3474  3533  3758  3790  3815
        3841  3883  3888  3906  3926  3999  4013  4015  4041  4092  4105  4120  4136  4283  4461
        4484  4509  4644  4646  4685  4995  5001  5108  5122  5130  5137  5160  5162
EMT      748
HALT     735   959  1019  4767  4841  4858
INC      975  1006  1739  1764  2400  4534  4754  4845  4980
INCB    4730  4735  4805  4809
JMP      736   740   933   967  1021  1473  1798  1802  1917  2010  2690  3304  3504  4088  4131
        4157  4170  4182  4183  4772  4856  5025  5065
JSR     1937  1941  1952  1956  1991  1998  2005  2620  2828  3043  3303  3499  3503  3898  4168
        4315  4333  4350  4468  4612  4626  4639  4764  5139  5145  5151
MOV      919   920   921   922   923   924   925   943   946   954   955   956   957   962   963
         970   972   977   978   979   986   997  1010  1014  1018  1027  1028  1033  1034  1035
        1036  1037  1038  1039  1040  1041  1049  1098  1099  1100  1101  1102  1103  1104  1105
        1106  1107  1108  1109  1113  1152  1153  1157  1161  1168  1183  1186  1187  1189  1193
        1197  1202  1206  1210  1214  1222  1224  1225  1235  1244  1248  1252  1260  1262  1263
        1272  1281  1285  1289  1297  1299  1300  1309  1318  1329  1335  1341  1350  1351  1358
        1365  1366  1373  1374  1385  1397  1400  1402  1409  1421  1438  1439  1440  1442  1449
        1453  1454  1465  1466  1467  1468  1469  1518  1547  1560  1574  1616  1617  1640  1641
        1648  1652  1653  1689  1717  1718  1726  1728  1729  1740  1741  1749  1750  1752  1753
        1765  1785  1794  1799  1800  1804  1806  1807  1817  1818  1824  1850  1861  1862  1863
        1867  1868  1875  1876  1884  1885  1892  1933  1934  1948  1949  1972  1974  1979  1982
        1984  1985  1986  1987  1993  2000  2001  2031  2034  2035  2040  2055  2085  2106  2110
        2115  2130  2162  2184  2185  2192  2196  2197  2235  2236  2247  2248  2289  2290  2297
        2301  2302  2338  2339  2349  2350  2390  2394  2396  2404  2405  2406  2410  2430  2436
        2437  2459  2461  2474  2476  2481  2483  2485  2486  2492  2493  2494  2508  2534  2560
        2589  2590  2591  2615  2622  2624  2625  2627  2640  2718  2720  2725  2727  2728  2729
```

```
            2740  2741  2742  2743  2744  2762  2787  2801  2802  2809  2811  2812  2827  2830  2831
            2833  2834  2836  2859  2896  2897  2898  2899  2910  2912  2936  2944  2946  2951  2953
            2954  2955  2961  2962  2963  2977  3003  3016  3017  3024  3026  3027  3042  3045  3046
            3048  3049  3051  3074  3121  3132  3143  3144  3148  3149  3150  3156  3157  3158  3159
            3160  3179  3204  3217  3218  3225  3227  3228  3243  3245  3246  3248  3270  3318  3329
            3338  3339  3343  3344  3345  3351  3352  3353  3354  3355  3373  3398  3411  3412  3419
            3421  3422  3437  3439  3440  3442  3464  3522  3523  3525  3526  3527  3532  3537  3538
            3557  3558  3566  3567  3587  3588  3597  3598  3637  3638  3645  3652  3653  3693  3730
            3732  3733  3734  3746  3748  3753  3755  3756  3757  3762  3763  3764  3765  3766  3780
            3806  3832  3861  3862  3863  3887  3897  3900  3902  3903  3905  3917  3985  3991  3992
            3993  3996  3997  3998  4002  4007  4008  4009  4010  4011  4012  4032  4089  4091  4097
            4098  4100  4101  4102  4113  4115  4116  4117  4135  4138  4140  4141  4142  4166  4191
            4195  4218  4219  4224  4225  4233  4234  4236  4237  4248  4253  4264  4266  4274  4313
            4316  4331  4334  4343  4344  4390  4391  4425  4432  4441  4448  4452  4469  4473  4474
            4486  4493  4500  4501  4511  4518  4580  4583  4584  4585  4593  4596  4597  4611  4617
            4625  4638  4649  4651  4656  4671  4672  4673  4676  4691  4697  4698  4699  4715  4716
            4718  4721  4731  4732  4736  4739  4758  4761  4781  4788  4789  4790  4791  4816  4817
            4818  4819  4831  4832  4833  4834  4835  4836  4837  4838  4839  4840  4843  4847  4848
            4849  4850  4851  4852  4853  4854  4868  4869  4870  4871  4872  4875  4891  4892  4893
            4894  4907  4908  4926  4937  4939  4941  4968  4969  4974  4990  4992  4998  5019  5022
            5028  5032  5036  5043  5048  5053  5058  5063  5070  5075  5080  5085  5091  5092  5096
            5098  5106  5116  5117  5119  5120  5124  5125  5128  5129  5132  5133  5143  5149  5153
            5154  5155  5157
MOVB        1322  1386  1390  1410  1414  1422  1426  1973  1994  2425  4681  4682  4687  4729  4760
            4783  4786  4813  4876  4913  4920
NOP         3534  3535  3954  3973  4153
RESET       1042  1179
ROL          949  1229  1267  1304  2401  2479  2723  2949  3301  3501  3595  3751  4281  4282  4290
            4291  4348  4459  4460  4465  4466  4600  4601  4607  4608  4797  4799  4801  4885  4886
            4887  4888  4889  4994  5093  5094  5100  5101  5102  5103  5104  5105  5136
ROLB        4798  4800  4802
ROR         4882  4883  4884
RTI         4216  4222  4230  4235  4242  4265  4284  4296  4320  4338  4394  4428  4433  4453  4462
            4497  4522  4700  4733  4740  4771  4820  4895  4927  4972  4975  5005  5007  5013
RTS         4193  4197  4352  4356  4358  4360  4362  4364  4366  4368  4370  4372  4374  4376  4378
            4380  4382  4384  4386  4529  4550  4563  4574  4652  4660  4985  4997  5111  5166
SEC          948  2478  2722  2948  3750
SUB         1634  1783  2626  2810  2835  3025  3050  3226  3247  3420  3441  3904  4759  4938
TRAP        4940  4943  4944  4945  4946  4947  4948  4949  4950  4951  4952  4953  4954  4955  4956
            4957  4958  4959  4960  4961  4962  4963  4964  4965  4966
TST          950  1060  1066  1072  1090  1119  1125  1237  1274  1311  1343  1508  1540  1935  1950
            2056  2163  2215  2266  2317  2364  2397  2422  2679  2682  2885  3101  3298  3492  3619
            3668  3714  4061  4108  4123  4150  4269  4305  4308  4326  4438  4526  4614  4628  4641
            4717  4737  4765  4981  5016
TSTB        4103  4118  4143  4677  4683  4688  4725  4803  4806  4911  4978
.ASCIZ       936   942   959   983   996  1018  1867  1889  1898  1978  1991  1998  2005  2202  2253
            2307  2354  2444  2912  3499  3657  3671  4131  4148  4166  4178  4191  4199  4205  4352
            4395  4396  4397  4398  4400  4402  4404  4406  4408  4410  4412  4414  4416  4417  4418
            4420  4758  4764  4856  4899  4919  5019  5022  5028  5032  5036  5043  5048  5053  5058
            5063  5070  5075  5080  5085  5091  5111  5143  5149  5165
.BLKB       4929
.BLKW       4822  5112  5113
.ENABL         1   718
.END        5167
.ENDC        778   920   926  1032  1096  1147  1165  1177  1203  1219  1231  1240  1256  1270  1277
            1293  1307  1314  1327  1333  1339  1346  1356  1363  1371  1382  1395  1407  1419  1434
```

```
                1463  1482  1660  1835  1922  1962  2015  2065  2172  2225  2277  2327  2374  2449  2694
                2920  3110  3309  3509  3577  3627  3679  3721  3977  4082  4163  4165  4166  4171  4713
                4741  4755  4757  4765  4773  4839  4847  4856  4857  4910  4930  4944  4945  4946  4947
                4948  4949  4950  4951  4952  4953  4954  4955  4956  4957  4958  4959  4960  4961  4962
                4963  4964  4965  4966  4967
.EVEN            936   942   959   983   996  1018  1867  1889  1898  1978  1991  1998  2005  2202  2253
                2307  2354  2444  2912  3499  3657  3671  4131  4148  4166  4178  4191  4212  4352  4758
                4764  4856  4899  4919  5019  5022  5028  5032  5036  5043  5048  5053  5058  5063  5070
                5075  5080  5085  5091  5111  5143  5149  5165
.IF              762   919   926  1029  1093  1144  1165  1174  1203  1219  1231  1240  1256  1270  1277
                1293  1307  1314  1327  1333  1339  1346  1356  1363  1371  1379  1395  1407  1419  1431
                1460  1479  1657  1832  1919  1959  2012  2062  2169  2222  2274  2324  2371  2446  2691
                2917  3107  3306  3506  3574  3624  3676  3718  3974  4079  4163  4164  4166  4170  4708
                4741  4751  4755  4764  4768  4839  4847  4855  4856  4909  4929  4943  4944  4945  4946
                4947  4948  4949  4950  4951  4952  4953  4954  4955  4956  4957  4958  4959  4960  4961
                4962  4963  4964  4965  4966
.IFF             920  4164  4170  4713  4751  4768  4856  4910  4930  4943  4944  4945  4946  4947  4948
                4949  4950  4951  4952  4953  4954  4955  4956  4957  4958  4959  4960  4961  4962  4963
                4964  4965  4966
.IIF             725   726   727   728   729   730   731   732   733   734  4172  4733  4737  4742  4774
                4775  4929
.IRP            4715  4718  4721  4788  4816  4833  4847  4870  4892  4943
.LIST              1   718   735   778   936   942   959   983   996  1018  1029  1033  1093  1097  1144
                1148  1165  1166  1174  1178  1203  1204  1219  1220  1231  1232  1240  1241  1256  1257
                1270  1271  1277  1278  1293  1294  1307  1308  1314  1315  1327  1328  1333  1334  1339
                1340  1346  1347  1356  1357  1363  1364  1371  1372  1379  1383  1395  1396  1407  1408
                1419  1420  1431  1435  1460  1464  1479  1483  1657  1661  1832  1836  1867  1889  1898
                1919  1923  1959  1963  1978  1991  1998  2005  2012  2016  2062  2066  2169  2173  2202
                2222  2226  2253  2274  2278  2307  2324  2328  2354  2371  2375  2444  2446  2450  2691
                2695  2912  2917  2921  3107  3111  3306  3310  3499  3506  3510  3574  3578  3624  3628
                3657  3671  3676  3680  3718  3722  3974  3978  4079  4083  4131  4148  4166  4178  4191
                4352  4758  4764  4856  4899  4919  4930  4943  4944  4945  4946  4947  4948  4949  4950
                4951  4952  4953  4954  4955  4956  4957  4958  4959  4960  4961  4962  4963  4964  4965
                4966  4967  5019  5022  5028  5032  5036  5043  5048  5053  5058  5063  5070  5075  5080
                5085  5091  5111  5143  5149  5165
.MACRO             1  4930
.MCALL           718   778  1029  1093  1144  1165  1174  1203  1219  1231  1240  1256  1270  1277  1293
                1307  1314  1327  1333  1339  1346  1356  1363  1371  1379  1395  1407  1419  1431  1460
                1479  1657  1832  1919  1959  2012  2062  2169  2222  2274  2324  2371  2446  2691  2917
                3107  3306  3506  3574  3624  3676  3718  3974  4079
.NLIST             1   718   735   778   936   942   959   983   996  1018  1029  1033  1093  1097  1144
                1148  1165  1166  1174  1178  1203  1204  1219  1220  1231  1232  1240  1241  1256  1257
                1270  1271  1277  1278  1293  1294  1307  1308  1314  1315  1327  1328  1333  1334  1339
                1340  1346  1347  1356  1357  1363  1364  1371  1372  1379  1383  1395  1396  1407  1408
                1419  1420  1431  1435  1460  1464  1479  1483  1657  1661  1832  1836  1867  1889  1898
                1919  1923  1959  1963  1978  1991  1998  2005  2012  2016  2062  2066  2169  2173  2202
                2222  2226  2253  2274  2278  2307  2324  2328  2354  2371  2375  2444  2446  2450  2691
                2695  2912  2917  2921  3107  3111  3306  3310  3499  3506  3510  3574  3578  3624  3628
                3657  3671  3676  3680  3718  3722  3974  3978  4079  4083  4131  4148  4166  4178  4191
                4352  4758  4764  4856  4899  4919  4930  4943  4944  4945  4946  4947  4948  4949  4950
                4951  4952  4953  4954  4955  4956  4957  4958  4959  4960  4961  4962  4963  4964  4965
                4966  4967  5019  5022  5028  5032  5036  5043  5048  5053  5058  5063  5070  5075  5080
                5085  5091  5111  5143  5149  5165
.PAGE            746   781  4701  4743  4775  4823  4863  4900  4930
.REM               1
.REPT            735
```

DERSCB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
.SBTTL   1029   1093   1144   1174   1379   1431   1460   1479   1657   1832   1919   1959   2012   2062   2169
         2222   2274   2324   2371   2446   2691   2917   3107   3306   3506   3574   3624   3676   3718   3974
         4079   4158   4661   4701   4743   4775   4823   4863   4900   4930
.TITLE    718


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DERSCB.SEQ/SOL/CRF/PAGNUM/NL:TOC/DS:ERFZ=SYSMAC.SML,DERSCB.P11
 RUN-TIME: 29 47 7 SECONDS
 RUN-TIME RATIO: 235/83=2.8
 CORE USED:  22K  (43 PAGES)
```