RH11/RS04

MAINT MODE DIAGNOSTIC

MD-11-DZRSD-C

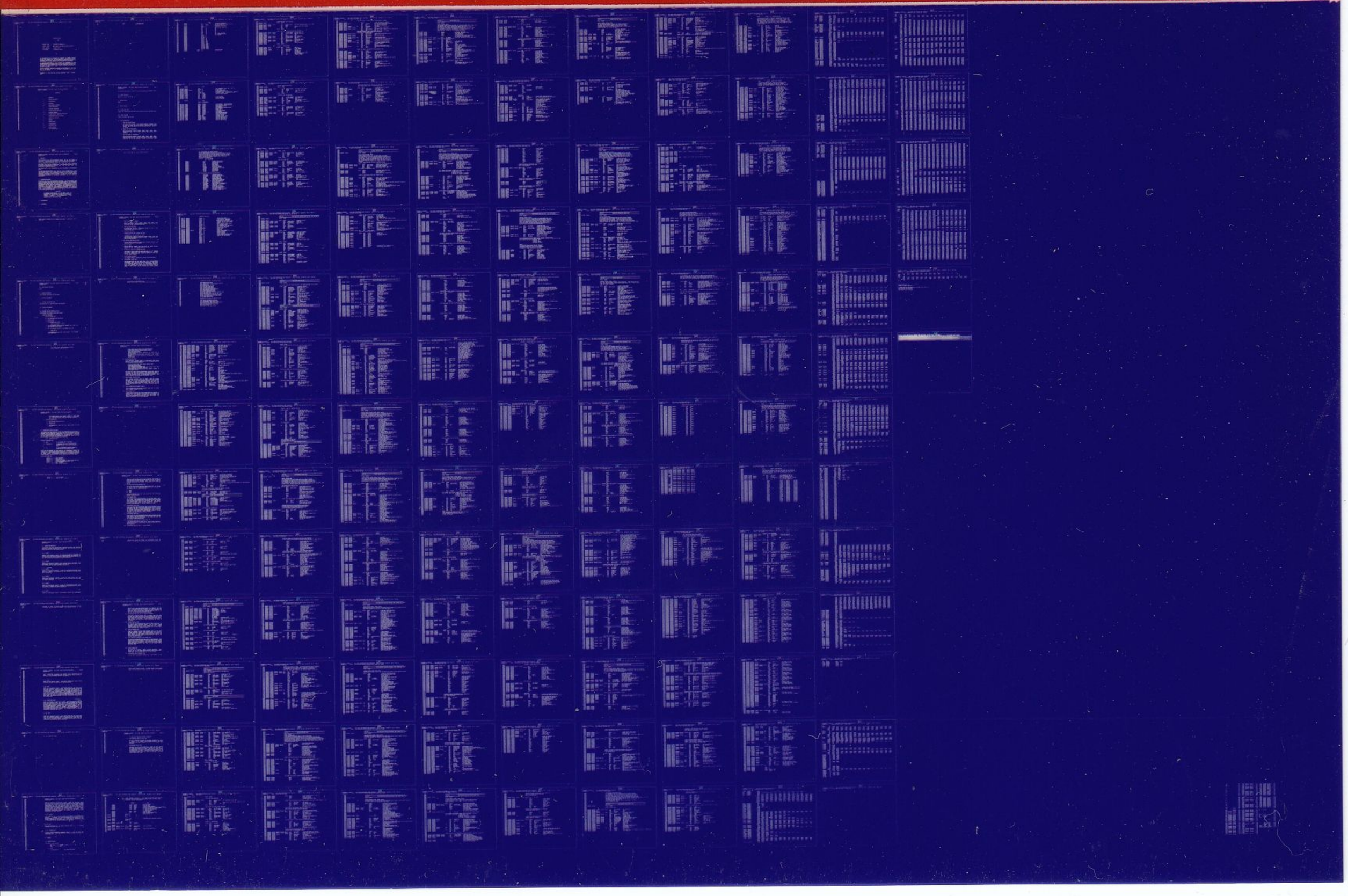
EP-DZRSD-C-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
                                .REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRSD-C-D
PRODUCT NAME:	RH11-RS04 MAINTENANCE MODE DIAGNOSTIC
DATE CREATED:	AUGUST 1976
MAINTAINER:	DIAGNOSTIC GROUP

CONTENTS

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 RS04 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 RS04 LOGIC IS REQUIRED TO UTILIZE THIS DIAGNOSTIC EFFECTIVELY. REFER TO SECTIONS 2 AND 3 OF THE "RS04 DECDISK SERVICE MANUAL" (DEC-00-HRS4A-A-D) FOR DESCRIPTIONS OF THE DRIVE LOGIC.

2. REQUIREMENTS

2.1 EQUIPMENT

```
                        PDP-11 WITH A MINIMUM OF 8K OF MEMORY AND AN RH11  CONTROLLER  WITH  A
                        RSD4 DISK.
```

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESSES

4.3 PROGRAM AND/C? 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 RSD4 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 RS04 DRIVES ON THE SYSTEM.

5. OPERATIONAL SWITCH SETTINGS

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC.176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(I.E.) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1.	<CR>	IF NO CHANGES ARE TO BE MADE.
2.	6 DIGITS 0-7	TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE; LAST DIGIT FOLLOWED BY <CR>
3.	↑U	TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

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 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

```
330                     REGISTER  TO  EQUAL  THE VALUE FOLLOWING THE MRCK INSTRUCTION.  IF THE
331                     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 THE TWO DATA BITS THAT ARE CURRENTLY BEING WRITTEN FROM THE DATA BUFFER IN CORE AND STORES ONE BIT IN A LOCATION CALLED NOWOD AND THE OTHER BIT IN LOCATION NOWEV. THE PREVIOUS CONTENST OF NOWOD AND NOWEV ARE STORED IN LASTOD AND LASTEV, RESPECITVELY. THIS INFORMATION IS USED BY THE CLKD1 AND CLKD2 ROUTINES TO DETERMINE THE CORRECT STATES OF THE MWDB (BIT 12) AND MWDT (BIT 14) IN BITS IN RSMR WHEN WRITING. THIS ROUTINE MAKES BITS 16 AND 17 OF EACH DATA WORD (RS04 WRITES 18 BIT WORDS) EQUAL ZERO. THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE XBIT INSTRUCTION.

5.1.9 CLKD1 AND CLKD2

TRAPS TO LOCATIONS ".CLKD1" AND ".CLKD2". THESE TWO ROUTINES USE THE DATA BITS RECEIVED FROM THE XBIT ROUTINE TO DETERMINE THE CORRECT STATES OF MWDB (BIT 12) AND MWDT (BIT 14) 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 TWO DATA BITS THAT ARE CURRENTLY BEING "READ" FROM THE DISK FROM THE INBUF DATA TABLE IN CORE AND STORES ONE BIT IN A LOCATION CALLED NOWOD AND THE OTHER BIT IN LOCATION NOWEV. THE PROGRAM THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE RBIT INSTRUCTION.

```
388
389                                 5.1.11  CLKR1 AND CLKR2
```

TRAPS TO LOCATIONS ".CLKR1" AND ".CLKR2". THESE TWO ROUTINES USING THE DATA BITS RECRIVED FROM THE RBIT ROUTINE SET AND CLEAR THE MRDB (BIT 2) AND MRDT (BIT 5) BITS 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 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.

5.1.13 HLT

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

5.1.14 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/RS04 REGISTERS.

```
446                             GOOD            = EXPECTED DATA.
447                             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

```
504                                     THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.
```

4. TEST "CLEAR BIT" IN RSCS2

   SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC RSDB, AND RSMR. SET CLEAR BIT IN RSCS2. NOW TEST ALL R/W BITS FOR 0 IN ALL THE ABOVE REGISTERS.

5. TEST DLT AND TRE BITS

   DO A READ FROM THE SILO. THIS SHOULD CAUSE A DLT AND A TRE ERROR BECAUSE THE SILO IS EMPTY.

6. CLEAR DLT AND TRE

   CLEAR BY SETTING TRE IN RSCS1 AND TEST.

7. 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.

8. TEST FOR 66 LOCATIONS IN SILO

   THIS IS DONE BY PUTTING A BINARY COUNT IN EVERY LOCATION AND CHECKING THE OUTPUT FOR 66 WORDS.

9. TEST DLT ERROR

   THIS IS DONE BY LOADING THE SILO WITH 67 WORDS WITHOUT READING ANY OUT. THIS SHOULD CAUSE DLT TO SET.

10. FLOAT A "1" AND A "0" THROUGH THE SILO

    LOAD THE SILO WITH A WORD OF ZEROS AND FLOAT A "1" THROUGH THE WORD. THEN LOAD THE SILO WITH A WORD OF ALL ONES AND FLOAT A "0" THROUGH THE WORD. CHECK THE OUTPUT OF THE SILO FOR THE CORRECT ANSWER.

11. TEST PROGRAM INTERRUPT

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

12. MAINTENANCE TIMING TEST

    THE FOLLOWING TEST ON THE RS04 DISK IS A SINGLE-STEPPED MAINTENANCE MODE TEST ON THE RS04 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.

13. 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 SHOULD INDICATE 7700 ON ANOTHER MAINTENANCE CLOCK.

14. 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.

15. 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.

16. 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.

```
619
620                                    17.  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.

18. 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).

19. 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.

20. MAINTENANCE WRITE TEST

THIS IS AN RS04 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.

21. MAINTENANCE READ TEST

THIS IS AN RS04 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.)

22. MAINTENANCE MODE DATA WRITE CHECK TEST

A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION. WITHIN THE RS04, A WRITE CHECK FUNCTION IS IDENTICAL TO A READ FUNCTION.

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

```
677                     THE RS04 DISK IS SET UP TO READ  (IN  MAINTENANCE  MODE)  ONE
678                     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.

24. 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.

25. 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.

26. 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).

27. 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).

28. 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.

29. MAINTENANCE MODE INTERRUPT TEST

IN THIS TEST THE INTERRUPT ENABLE (I.E.,) BIT IS SET. A TWO

```
735                                     SECTOR WRITE COMMAND IS GIVEN.  AN "RMR" ERROR IS THEN CAUSED
736                                     WHILE THE FIRST SECTOR IS BEING WRITTEN.  WHEN  THE  FUNCTION
```

IS COMPLETED, THE DRIVE SHOULD INTERRUPT.

30. 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.

31. 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-DZRSD-C      RS11-RS04 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
                                        SW13=   20000           ;INHIBIT ERROR TYPEOUTS
                                        SW12=   10000           ;TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
                                        SW11=   4000            ;RUN MAINTENANCE MODE VERIFY TEST
                                        SW10=   2000            ;0 - BELL ON PASS COMPLETE
                                                                ;1 - BELL ON ERROR
                                        SW9=    1000            ;LOOP ON ERROR
                                        SW8=    400             ;LOOP ON TEST IN SW<7:0>
                                .=      0                       ;TRAP CATCHER FROM 0 - 776
                                .=      46                      ;HOOKS FOR ACT 11
000046  021634                  $ENDAD
                                .=      52
000052  040000                  BIT14

                                .=      174                     ;SOFTWARE SWITCK REGISTER LOCATION
000174  000000                  DISPREG:0
000176  000000                  SWREG:  0

                                .=      200
000200  000137  000232          JMP     @#BEGIN1

                                .=      220
000220  052737  000100  001144  BIS     #BIT6,FLAG2     ;TEST ALL DRIVES
000226  000137  001234          BEGIN2: JMP     @#BEGIN

000232  042737  000100  001144  BEGIN1: BIC     #BIT6,FLAG2     ;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
        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
                                                
001000  000000          ICNT:   0               ;LH = ITERATION COUNT  ;RH = TEST NO.
001002  000000          ERRORS: 0               ;ERROR COUNT
001004  000000  000000  PCNT:   0,0             ;2 WORD PASS COUNT
001010  000000          LAD:    0               ;LOOP ADDRESS FOR SCOPE
001012  000000          HLTADR: 0               ;ADDRESS OF LAST HLT INSTRUCTION  EXECUTED
001014  001000          FILCHR: 1000            ;FILCHR=0 (CHAR)  ;FILCHR+1=2 (COUNT)
001016  177564          TPS:    177564          ;OUTPUT STATUS REGISTER
001020  177560          TKS:    177560
001022  177562          TKB:    177562
001024  177566          TPB:    177566          ;OUTPUT BUFFER
001026  177570          SWR:    177570          ;SWITCH REGISTER
001030  177570          DISPLAY:177570          ;DISPLAY REGISTER
                        
                        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             ;INTERUPT 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:   MLT !CS1,AS,BA
                                ;           MLT !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                LSTEV:  0       ;LAST EVEN BIT TRANSFERED
001150  000000                LSTOD:  0       ;LAST ODD BIT TRANSFERED
001152  000000                NOWEV:  0       ;PRESENT EVEN BIT BEING XFERED
001154  000000                NOWOD:  0       ;PRESENT ODD BIT BEING XFERED
001156  000000                RSD:    0       ;SAME
001160  000000                UNNUM:  0       ;UNIT CURRENTLY BEING TESTED
001162  000000                UNITSV: 0       ;SET BIT=UNIT ON BUS
001164  000000                UNCMP:  0       ;FOR COMPARING FOR # OF DEVICE
001166  000000                ONCEE:  0       ;DID WE TEST ANY DRIVES
001170  000000                TIMSV:  0       ;SAVE LOC FOR TIME
        172100                MPRD=172100     ;PARITY REG
001172  000000                SAVEE:  0       ;WORK LOC
001174  000000  000000        MCCNT:  0,0     ;MAINT CLOCK COUNT
001200  000000                WCRC:   0       ;WORK LOC FOR CREATING CRC WORD
001202  000000                REPT:   0       ;REPEAT COUNTER
001204  000000                REPT1:  0       ;REPEAT COUNTER
001206  000000                CLKCNT: 0       ;CLOCK COUNTER FOR EACH WORD
001210  000000                INBIT:  0       ;USED IN CRC CAL ROUTINE
001212  000000                WK15:   0       ;USED IN CRC CAL ROUTINE
001214  000000                WORK:   0
001216  000000                WORK0:  0
001220  000000                WORK1:  0
001222  000000                WORK2:  0
001224  000000                WORK3:  0
001226  000000                WORK4:  0
001230  000000                WORK5:  0
001232  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
                        ;BIT4 TEST ONLY ONE DRIVE
                        ;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 IN CRC TEST
                        ;BIT6 FLAG TO TEST ALL DRIVES
```

```
  968  001234  012706  000500          BEGIN:  MOV     #500,SP         ;SET STACK TO  *** 500 ***
  969  001240  012737  025442  000024          MOV     #.POWER,@#24    ;SET UP PF VECTOR
  970  001246  012737  000340  000026          MOV     #340,@#26       ;LOCK OUT THE WORLD
  971  001254  012737  025070  000030          MOV     #.HLT,@#30      ;SET EMT VECTOR
  972  001262  012737  000340  000032          MOV     #340,@#32       ;LOCK UP
  973  001270  012737  026076  000034          MOV     #.TRAP,@#34     ;SET TRAP VECTOR
  974  001276  012737  000340  000036          MOV     #340,@#36       ;LOCK UP
  975  001304  005037  001000                  CLR     ICNT            ;INIT ICNT
  976  001310  005037  001010                  CLR     LAD             ;INIT LAD
  977  001314  104454                          SUSWR                   ;SIZE FOR HDWR SWR
  978  001316  042737  177677  001144          BIC     #177677,FLAG2
  979  001324  042737  153777  001166          BIC     #153777,ONCEE
  980  001332  032737  000100  001144          BIT     #BIT6,FLAG2     ;TEST ALL DRIVES?
  981  001340  001402                          BEQ     5$              ;ASK
  982  001342  000137  001674                  JMP     @#MULTII
  983  001346                          5$:
  984  001346  104402  001352                  TYPE    ,.+2            ;.ASCIZ <15><12>"TEST ALL DRIVES? (Y OR N) "
  985  001410  104412                          RDLIN
  986  001412  122737  000131  026056          CMPB    #'Y,INPUT       ;TEST FOR YES
  987  001420  001525                          BEQ     MULTII          ;YES
  988  001422  052737  000020  001166          BIS     #BIT4,ONCEE     ;SET TEST ONLY ONE DRIVE FLAG
  989  001430                          1$:
  990  001430  104402  001434                  TYPE    ,.+2            ;.ASCIZ "TYPE UNIT #"
  991  001450  104410                          RDOCT
  992  001452  012604                          MOV     (6)+,R4         ;GET NUMBER
  993  001454  022704  000010                  CMP     #10,R4          ;CORRECT #
  994  001460  101763                          BLOS    1$              ;NO
  995  001462  010437  001160                  MOV     R4,UNNUM        ;SET UNIT #
  996  001466  005002                          CLR     R2              ;CLEAR WORK AREA
  997  001470  000261                          SEC                     ;SET CARRY
  998  001472  006102                  2$:     ROL     R2              ;SET WORK BIT
  999  001474  005704                          TST     R4              ;IS THIS BIT CORRESPOND WITH CORRECT DRIVE #
 1000  001476  001402                          BEQ     3$              ;YES
 1001  001500  005304                          DEC     R4              ;NO TRY AGAIN
 1002  001502  000773                          BR      2$              ;TEST AGAIN
 1003  001504  010237  001162          3$:     MOV     R2,UNITSV       ;SET DRIVE BIT IN UNITSV
 1004  001510  010237  001164                  MOV     R2,UNCMP        ;SET UNIT COMPARE
 1005  001514  013777  001160  177360          MOV     UNNUM,@RSCS2    ;LOAD DRIVE
 1006  001522  012777  177777  177364          MOV     #-1,@RSER       ;LOAD ERRORS
 1007  001530  104402  001534                  TYPE    ,.+2            ;.ASCIZ "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD BE ON
 1008  001644  000000                          HALT                    ;WAIT FOR LIGHTS TO BE CHECKED
 1009  001646  023777  001162  177242          CMP     UNITSV,@RSAS    ;DID CORRECT ATA SET
 1010  001654  001405                          BEQ     4$
 1011  001656  017700  177234                  MOV     @RSAS,BAD       ;GET RSAS
 1012  001662  013701  001162                  MOV     UNITSV,GOOD     ;GET CORRECT AND
 1013  001666  104000                          HLT                     ;RSAS=BAD GOOD=CORRECTIONS
 1014                                                                  ;ATA BIT SHOULD SET FOR ERRORS
 1015                                                                  ;WERE SET IN RSER
 1016  001670  000137  002324          4$:     JMP     NOWGO           ;START TESTING
```

```
1017                                    ;NOW TEST FOR DRIVES
1018
1019  001674  012701  000010      MULTII: MOV     #8.,R1          ;PUT 8 INTO R1 FOR COUNT
1020  001700  005077  177176              CLR     @RSCS2          ;SET DEVICE TO ZERO
1021  001704  012777  177777  177202 TRY:    MOV     #-1,@RSER       ;CAUSE AN ERROR +SETS BIT IN RSAS REG
1022  001712  005301                      DEC     R1              ;DO A MAXIMUM OF 8 TIMES
1023  001714  001403                      BEQ     DVNUM           ;TESTED FOR ALL DRIVES GET OUT
1024  001716  005277  177160              INC     @RSCS2          ;INCREMENT DRIVE UNIT
1025  001722  000770                      BR      TRY             ;REPEAT FOR NEXT DRIVE
1026  001724  017737  177166  001162 DVNUM:  MOV     @RSAS,UNITSV    ;SAVE
1027  001732  012737  000401  001164       MOV     #401,UNCMP      ;SETUP TO CMP WITH UNITSV
1028  001740  012737  000000  001160       MOV     #0,UNNUM        ;PUT 0 INTO UNIT NO.
1029  001746  032777  020000  177052       BIT     #BIT13,@SWR     ;INHIBIT TYPE OUT?
1030  001754  001015                      BNE     STTEST          ;YES
1031  001756  104402  001762              TYPE    ,.+2            ;.ASCIZ <15><12>"TESTING UNIT "
1032  002002  042737  100000  001166       BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
1033  002010  033737  001164  001162 STTEST: BIT     UNCMP,UNITSV    ;IS THIS DRIVE ON THE SYSTEM
1034  002016  001440                      BEQ     TRYNX           ;NO
1035  002020  013777  001160  177054       MOV     UNNUM,@RSCS2    ;YES PUT UNIT # INTO CS2
1036  002026  022777  000002  177072 3$:    CMP     #2,@RSDT        ;IS THIS A RS04?
1037  002034  001404                      BEQ     1$              ;YES
1038  002036  022777  000003  177062       CMP     #3,@RSDT        ;IS IT A RS04?
1039  002044  001025                      BNE     TRYNX           ;GET A NEW NUMBER
1040  002046  032777  020000  176752 1$:    BIT     #BIT13,@SWR     ;INHIBIT TYPE OUT?
1041  002054  001020                      BNE     4$              ;YES
1042  002056  032737  100000  001166       BIT     #BIT15,ONCEE    ;ANY ERRORS?
1043  002064  001404                      BEQ     5$              ;NO
1044  002066  104402  002072              TYPE    ,.+2            ;.ASCIZ <15><12><12>
1045  002076                        5$:
1046  002076  013746  001160              MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
1047  002102  104406                      TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
1048  002104  104402  000040              TYPE    ,40             ;TYPE SPACE
1049  002110  042737  100000  001166       BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
1050  002116  000502                4$:    BR      NOWGO           ;NOW TEST
1051  002120  032737  000020  001166 TRYNX:  BIT     #BIT4,ONCEE     ;MULTI DRIVE
1052  002126  001074                      BNE     DONEE           ;NO
1053  002130  006337  001164       1$:    ASL     UNCMP           ;CHECK NEXT BIT FOR DRIVE
1054  002134  103403                      BCS     CHCKDV          ;DID WE TEST ANY REG?
1055  002136  005237  001160              INC     UNNUM           ;INC UNIT #
1056  002142  000722                      BR      STTEST          ;CHECK FOR NEXT DRIVE
```

```
 1057  002144  032737  000001  001166  CHCKDV: BIT     #BIT0,ONCEE     ;DID WE TEST ANY DRIVES?
 1058  002152  001062                          BNE     DONEE           ;YES WE DID TEST A DRIVE
 1059  002154  012737  100000  001164          MOV     #100000,UNCMP   ;NO DRIVES TESTED, COULD NOT SET
 1060  002162  005037  001160                  CLR     UNNUM           ;ANY AS BITS, THUS DEFAULTS TO
 1061  002166  032777  020000  176632          BIT     #BIT13,@SWR     ;INHIBIT TYPE OUT?
 1062  002174  001050                          BNE     4$              ;YES
 1063  002176  013746  001160                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
 1064  002202  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
 1065  002204  104402  000040                  TYPE    ,40             ;TYPE SPACE
 1066  002210  104402  002214                  TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
 1067  002306  012737  000001  001164          MOV     #1,UNCMP
 1068  002314  000000                          HALT                    ;WAIT
 1069  002316  000402                  4$:     BR      NOWGO           ;TEST DRIVE 0
 1070  002320  000137  021570          DONEE:  JMP     DONE            ;GET OUT
 1071
 1072                                  ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
 1073                                  ;TO CLEAR ALL THE RH AND RS REGISTERS
 1074
 1075  002324  052737  000001  001166  NOWGO:  BIS     #BIT0,ONCEE     ;SET FOUND DRIVE FLAG
 1076  002332  013737  025066  001170          MOV     TIMES,TIMSV     ;SAVE TIME
 1077  002340  012737  000001  025066          MOV     #1,TIMES        ;ONLY TEST ONCE
 1078                                  ;*************************************************************************
 1079                                  ;TEST 1         RESET TEST FOR REGISTERS
 1080                                  ;*************************************************************************
 1081  002346  104400                  TST1:   SCOPE
 1082  002350  012737  000340  177776          MOV     #340,@#PS       ;LOCK OUT INTERUPTS
 1083  002356  013777  001160  176516          MOV     UNNUM,@RSCS2    ;LOAD UNIT #
 1084  002364  012777  177776  176506          MOV     #177776,@RSCS1  ;SET ALL
 1085  002372  012777  177777  176506          MOV     #177777,@RSBA   ;POSSIBLE R/W
 1086  002400  012777  177777  176502          MOV     #177777,@RSDA   ;BITS IN THESE REGISTERS
 1087  002406  012777  177777  176500          MOV     #177777,@RSER
 1088  002414  012777  177777  176502          MOV     #177777,@RSMR
 1089  002422  012777  177777  176454          MOV     #177777,@RSWC
 1090  002430  012777  177737  176444          MOV     #177737,@RSCS2
 1091  002436  000005                          RESET                   ;CLEAR ALL BITS IN ALL REG.
 1092
 1093                                  ;TEST RSCS2 FOR CLEARED BITS
 1094
 1095  002440  022777  000100  176434          CMP     #100,@RSCS2     ;DID THESE BITS GET CLEARED?
 1096  002446  001401                          BEQ     .+4             ;YES
 1097  002450  104200                          HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
 1098  002452  013777  001160  176422          MOV     UNNUM,@RSCS2    ;PUT # OF UNIT IN TEST IN CS2
 1099  002460  022777  010600  176424          CMP     #10600,@RSDS    ;IS DPR AND MOL SET?
 1100  002466  001401                          BEQ     .+4             ;YES
 1101  002470  104040                          HLT     !DS             ;NO WHY NOT?
 1102
 1103                                  ;TEST CONTROL AND STATUS REG 1
 1104  002472  022777  004200  176400          CMP     #4200,@RSCS1    ;DID THE READY BIT SET?
 1105  002500  001401                          BEQ     .+4             ;YES
 1106  002502  104001                          HLT     !CS1            ;READY SHOULD BE SET
```

```
1107                                    ;TEST BUS ADDRESS REGISTER
1108
1109  002504  005777  176376                  TST     @RSBA           ;IS BA REG. CLEARED
1110  002510  001401                          BEQ     .+4             ;YES
1111  002512  104020                          HLT     !BA             ;SHOULD BE 0
1112
1113                                    ;TEST DISK ADDRESS REGISTER
1114
1115  002514  005777  176370                  TST     @RSDA           ;IS DA CLEARED
1116  002520  001401                          BEQ     .+4             ;YES
1117  002522  104004                          HLT     !DA             ;SHOULD BE 0
1118
1119                                    ;TEST ERROR REG RSER
1120
1121  002524  005777  176364                  TST     @RSER           ;DID RSER CLEAR?
1122  002530  001401                          BEQ     .+4             ;YES
1123  002532  104002                          HLT     !ER             ;BITS(157015) SHOULD BE CLEARED
1124
1125                                    ;TEST RS MAINTENANCE REGISTER
1126
1127  002534  032777  000077  176362          BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
1128  002542  001401                          BEQ     .+4             ;YES
1129  002544  104220                          HLT     !MR             ;BITS(77) SHOULD BE 0
1130
1131                                    ;TEST WC REG IT SHOULD NOT CHANGE
1132
1133  002546  022777  177777  176330          CMP     #177777,@RSWC   ;DID IT CHANGE?
1134  002554  001401                          BEQ     .+4             ;NO
1135  002556  104010                          HLT     !WC             ;RESET SHOULD NOT MODIFY RSWC
1136
1137                                    ;TEST RSAS
1138
1139  002560  005777  176332                  TST     @RSAS           ;IS REG CLEAR
1140  002564  001401                          BEQ     .+4             ;YES
1141  002566  104100                          HLT     !AS             ;NO
```

```
1142                                    ;*************************************************************************
1143                                    ;TEST 2          TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
1144                                    ;*************************************************************************
1145  002570  104400                    TST2:   SCOPE
1146
1147  002572  012737  000340  177776    TTAGG:  MOV     #340,@#PS       ;LOCK OUT INTERRUPTS
1148  002600  013777  001160  176274            MOV     UNNUM,@RSCS2
1149  002606  012777  043576  176264            MOV     #43576,@RSCS1   ;SET ALL
1150  002614  012777  177777  176264            MOV     #177777,@RSBA   ;POSSIBLE
1151  002622  012777  177777  176260            MOV     #177777,@RSDA   ;REGISTERS
1152  002630  012777  177017  176256            MOV     #177017,@RSER
1153  002636  012777  177777  176256            MOV     #177777,@RSDB
1154  002644  012777  177777  176232            MOV     #177777,@RSWC
1155  002652  012777  020417  176222            MOV     #20417,@RSCS2
1156  002660  012777  000071  176236            MOV     #71,@RSMR
1157  002666  012777  000040  176206            MOV     #40,@RSCS2      ;CLEAR ALL BITS
1158  002674  022777  000100  176200            CMP     #100,@RSCS2     ;DID THE RIGHT BITS CLEAR?
1159  002702  001401                            BEQ     .+4             ;YES
1160  002704  104200                            HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
1161  002706  013777  001160  176166            MOV     UNNUM,@RSCS2    ;GET DRIVE NUMBER
1162  002714  032777  173577  176156            BIT     #173577,@RSCS1  ;DID ALL BITS GET CLEARED
1163  002722  001401                            BEQ     .+4             ;YES
1164  002724  104001                            HLT     !CS1            ;NO, ALL BITS SHOULD BE 0
1165                                    ;TEST BUS ADDRESS REGISTER
1166
1167  002726  005777  176154                    TST     @RSBA           ;IS BA REG. CLEARED
1168  002732  001401                            BEQ     .+4             ;YES
1169  002734  104020                            HLT     !BA             ;SHOULD BE 0
1170
1171                                    ;TEST DISK ADDRESS REGISTER
1172
1173  002736  005777  176146                    TST     @RSDA           ;IS DA CLEARED
1174  002742  001401                            BEQ     .+4             ;YES
1175  002744  104020                            HLT     !BA             ;SHOULD BE 0
1176
1177                                    ;TEST ERROR REG RSER
1178
1179  002746  032777  177777  176140            BIT     #177777,@RSER   ;DID THESE BITS GET CLEARED
1180  002754  001401                            BEQ     .+4             ;YES
1181  002756  104002                            HLT     !ER             ;BITS(157015) SHOULD BE CLEARED
1182
1183                                    ;TEST RS MAINTENANCE REGISTER
1184  002760  032777  000077  176136            BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
1185  002766  001401                            BEQ     .+4             ;YES
1186  002770  104220                            HLT     !MR             ;BITS(77) SHOULD BE 0
1187
1188                                    ;TEST WC REG. IT SHOULD NOT CHANGE
1189  002772  022777  177777  176104            CMP     #177777,@RSWC   ;DID WC CHANGE
1190  003000  001401                            BEQ     .+4             ;NO
1191  003002  104010                            HLT     !WC             ;WHY DID IT CHANGE?
```

```
1192                                ;*****************************************************************************
1193                                ;TEST 3          SET AND CLEAR ALL REGISTERS
1194                                ;*****************************************************************************
1195  003004  104400                TST3:   SCOPE
1196                                ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
1197                                ;BITS 7,6,5,4,3,2&1
1198
1199  003006  104414                        CLRDK                   ;CLEAR ALL RS REG
1200  003010  013737  001170  025066        MOV     TIMSV,TIMES     ;GET TIME
1201  003016  012777  003576  176054        MOV     #3576,@RSCS1    ;SET DISK FUNCTION BITS
1202  003024  022777  007776  176046        CMP     #7776,@RSCS1    ;ARE THESE BITS SET?
1203  003032  001401                       BEQ     .+4             ;NO
1204  003034  104001                       HLT     !CS1            ;SHOULD = 3776
1205  003036  012777  002524  176034        MOV     #2524,@RSCS1    ;SET THESE BITS
1206  003044  022777  006724  176026        CMP     #6724,@RSCS1    ;DID THEY SET
1207  003052  001401                       BEQ     .+4             ;YES
1208  003054  104001                       HLT     !CS1            ;SHOULD BE 2725
1209  003056  012777  001052  176014        MOV     #1052,@RSCS1    ;SET THESE BITS
1210  003064  022777  005252  176006        CMP     #5252,@RSCS1    ;ARE THEY =?
1211  003072  001401                       BEQ     .+4             ;YES
1212  003074  104001                       HLT     !CS1            ;SHOULD = 1252
1213  003076  104400                TST4:   SCOPE
1214                                ;CLEAR THE FUNCTION BITS
1215
1216  003100  012777  043576  175772        MOV     #43576,@RSCS1   ;SET DISK FUNCTION BITS
1217  003106  005077  175766               CLR     @RSCS1
1218  003112  022777  004200  175760        CMP     #4200,@RSCS1    ;IS THE READY BIT SET
1219  003120  001401                       BEQ     .+4             ;YES
1220  003122  104001                       HLT     !CS1            ;RSCS1 SHOULD = 4200
1221
1222                                ;*****************************************************************************
1223                                ;TEST 5          TEST RSCS2
1224                                ;*****************************************************************************
1225  003124  104400                TST5:   SCOPE
1226
1227  003126  000005                        RESET                   ;CLEAR WORLD
1228  003130  022777  000100  175744        CMP     #100,@RSCS2     ;DID THEY CLEAR?
1229  003136  001401                       BEQ     .+4             ;YES
1230  003140  104200                       HLT     !CS2            ;NO
1231  003142  012777  021037  175732        MOV     #21037,@RSCS2   ;SET BITS 21017
1232  003150  022777  021137  175724        CMP     #21137,@RSCS2   ;DID THESE BITS GET SET
1233  003156  001405                       BEQ     1$              ;YES
1234  003160  017700  175716               MOV     @RSCS2,BAD
1235  003164  012701  021137               MOV     #21137,GOOD     ;WHAT CS2 SHOULD =
1236  003170  104000                       HLT                     ;CS2 = BAD GOOD = CORRECT ANS
```

```
1237  003172  012777  020025  175702  1$:     MOV     #20025,@RSCS2   ;SET THESE BITS
1238  003200  022777  020125  175674          CMP     #20125,@RSCS2   ;DID THESE BITS GET SET
1239  003206  001401                          BEQ     .+4             ;YES
1240  003210  104200                          HLT     !CS2            ;NO,CS2 SHOULD = 20125
1241  003212  012777  000012  175662          MOV     #12,@RSCS2      ;LOAD THESE BITS
1242  003220  022777  000112  175654          CMP     #112,@RSCS2     ;DID THESE BITS GET SET IN CS2
1243  003226  001401                          BEQ     .+4             ;YES
1244  003230  104200                          HLT     !CS2            ;BAD = CS2 GOOD = CORRECT ANS
1245  003232  012777  177777  175642          MOV     #-1,@RSCS2      ;SET BITS
1246  003240  005077  175636                  CLR     @RSCS2          ;CLEAR THEM
1247  003244  022777  000100  175630          CMP     #100,@RSCS2     ;DID CLEAR WORK
1248  003252  001401                          BEQ     .+4             ;YES
1249  003254  104200                          HLT     !CS2            ;R/W BITS DID NOT CLEAR
1250  003256  013777  001160  175616          MOV     UNNUM,@RSCS2    ;GET UNIT #
1251  003264  104400                  TST6:   SCOPE
1252                                  ;CAN WE SET ALL THE RSBA BITS
1253
1254  003266  012777  177777  175612          MOV     #177777,@RSBA   ;SET THE BITS
1255  003274  022777  177776  175604          CMP     #177776,@RSBA   ;DID THEY SET
1256  003302  001401                          BEQ     .+4             ;YES
1257  003304  104020                          HLT     !BA             ;BITS 17776 SHOULD BE SET
1258  003306  012777  125252  175572          MOV     #125252,@RSBA   ;SET THESE BITS
1259  003314  022777  125252  175564          CMP     #125252,@RSBA   ;ARE THEY =
1260  003322  001401                          BEQ     .+4             ;YES
1261  003324  104020                          HLT     !BA             ;SHOULD BE 125252
1262  003326  012777  052524  175552          MOV     #52524,@RSBA    ;SET THESE BITS
1263  003334  022777  052524  175544          CMP     #52524,@RSBA    ;ARE THEY =
1264  003342  001401                          BEQ     .+4             ;YES
1265  003344  104020                          HLT     !BA             ;SHOULD BE 52524
1266
1267  003346  104400                  TST7:   SCOPE
1268                                  ;FLOAT A 1 THROUGH RSBA
1269
1270  003350  012701  000002          FLOTBA: MOV     #2,GOOD         ;GET A 2
1271  003354  000241                          CLC                     ;CLEAR CARRY
1272  003356  010177  175524          1$:     MOV     GOOD,@RSBA      ;FLOAT NUMBER
1273  003362  017700  175520                  MOV     @RSBA,BAD       ;GET BA
1274  003366  020100                          CMP     GOOD,BAD        ;COMPARE BA
1275  003370  001401                          BEQ     .+4             ;BA CORRECT
1276  003372  104000                          HLT                     ;BAD=BA GOOD=CORRECT ANS
1277  003374  006101                          ROL     GOOD            ;ROTATE NUMBER
1278  003376  103367                          BCC     1$              ;LOOP TILL DONE
```

```
1279  003400  104400                    TST10:  SCOPE
1280
1281                                    ;CLEAR THE RSBA REGISTER
1282
1283  003402  012777  177777  175476            MOV     #177777,@RSBA   ;SET RSBA EQUAL TO ALL ONES
1284  003410  005077  175472                    CLR     @RSBA
1285  003414  005777  175466                    TST     @RSBA           ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1286  003420  001401                          BEQ     .+4             ;YES
1287  003422  104020                          HLT     !BA             ;NO
1288  003424  104400                    TST11:  SCOPE
1289
1290                                    ;CAN WE SET ALL BITS IN RSWC REGISTER
1291
1292  003426  012777  177777  175450            MOV     #177777,@RSWC   ;SET WC BITS
1293  003434  022777  177777  175442            CMP     #177777,@RSWC   ;ARE ALL BITS SET
1294  003442  001401                          BEQ     .+4             ;YES
1295  003444  104010                          HLT     !WC             ;NO
1296  003446  012777  125252  175430            MOV     #125252,@RSWC   ;SET THESE BITS
1297  003454  022777  125252  175422            CMP     #125252,@RSWC   ;ARE THEY =
1298  003462  001401                          BEQ     .+4             ;YES
1299  003464  104010                          HLT     !WC             ;SHOULD BE 125252
1300  003466  012777  052525  175410            MOV     #52525,@RSWC    ;SET THESE BITS
1301  003474  022777  052525  175402            CMP     #52525,@RSWC    ;ARE THEY =
1302  003502  001401                          BEQ     .+4             ;YES
1303  003504  104010                          HLT     !WC             ;SHOULD BE 152525
1304  003506  104400                    TST12:  SCOPE
1305
1306                                    ;FLOAT A 1 THROUGH RSWC
1307
1308  003510  012701  000001          FLOTWC: MOV     #1,GOOD         ;GET A 1
1309  003514  000241                          CLC                     ;CLEAR CARRY
1310  003516  010177  175362          1$:     MOV     GOOD,@RSWC      ;FLOAT NUMBER
1311  003522  017700  175356                  MOV     @RSWC,BAD       ;GET WC
1312  003526  020100                          CMP     GOOD,BAD        ;COMPARE WC
1313  003530  001401                          BEQ     .+4             ;WC CORRECT
1314  003532  104000                          HLT                     ;BAD=WC GOOD=CORRECT ANS
1315  003534  006101                          ROL     GOOD            ;ROTATE NUMBER
1316  003536  103367                          BCC     1$              ;LOOP TILL DONE
```

```
1317                                        ;CLEAR THE WORD COUNT REGISTER
1318  003540  104400                        TST13:  SCOPE
1319
1320  003542  012777  177777  175334              MOV     #177777,@RSWC   ;SET RSWC REGISTER EQUAL TO ALL ONES
1321  003550  005077  175330                    CLR     @RSWC
1322  003554  005777  175324                    TST     @RSWC           ;DID ALL BITS GET CLEARED
1323  003560  001401                            BEQ     .+4             ;YES
1324  003562  104010                            HLT     !WC             ;NO
1325  003564  104400                        TST14:  SCOPE
1326
1327                                        ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
1328
1329  003566  012777  177777  175314              MOV     #177777,@RSDA   ;SET ALL BITS
1330  003574  022777  177777  175306              CMP     #177777,@RSDA   ;ARE THE BITS SET
1331  003602  001401                            BEQ     .+4             ;YES
1332  003604  104004                            HLT     !DA             ;NO
1333  003606  012777  125252  175274              MOV     #125252,@RSDA   ;SET THESE BITS
1334  003614  022777  125252  175266              CMP     #125252,@RSDA   ;ARE THEY =
1335  003622  001401                            BEQ     .+4             ;YES
1336  003624  104004                            HLT     !DA             ;SHOULD BE 125252
1337  003626  012777  052525  175254              MOV     #52525,@RSDA    ;SET THESE BITS
1338  003634  022777  052525  175246              CMP     #52525,@RSDA    ;ARE THEY =
1339  003642  001401                            BEQ     .+4             ;YES
1340  003644  104004                            HLT     !DA             ;SHOULD BE 52525
1341  003646  104400                        TST15:  SCOPE
1342
1343                                        ;FLOAT A 1 THROUGH RSDA
1344
1345  003650  012701  000001                FLOTDA: MOV     #1,GOOD         ;GET A 1
1346  003654  000241                            CLC                     ;CLEAR CARRY
1347  003656  010177  175226                1$:     MOV     GOOD,@RSDA      ;FLOAT NUMBER
1348  003662  017700  175222                    MOV     @RSDA,BAD       ;GET DA
1349  003666  020100                            CMP     GOOD,BAD        ;COMPARE DA
1350  003670  001401                            BEQ     .+4             ;DA CORRECT
1351  003672  104000                            HLT                     ;BAD=DA GOOD=CORRECT ANS
1352  003674  006101                            ROL     GOOD            ;ROTATE NUMBER
1353  003676  103367                            BCC     1$              ;LOOP TILL DONE
```

```
1354
1355  003700  104400                        ;CAN WE CLEAR THE RSDA REG.
                                  TST16:  SCOPE
1356
1357  003702  012777  177777  175200        MOV     #177777,@RSDA   ;SET RSDA TO ALL ONES
1358  003710  005077  175174                CLR     @RSDA
1359  003714  005777  175170                TST     @RSDA           ;TEST FOR ZERO RSDA
1360  003720  001401                        BEQ     .+4             ;YES
1361  003722  104004                        HLT     !DA             ;ANS SHOULD BE 0
1362  003724  104400                TST17:  SCOPE
1363
1364                                  ;SET AND CLEAR THE RSER REG.
1365
1366  003726  012777  177017  175160        MOV     #177017,@RSER   ;SET THESE BITS
1367  003734  022777  177017  175152        CMP     #177017,@RSER   ;DID THEY SET
1368  003742  001401                        BEQ     .+4             ;YES
1369  003744  104002                        HLT     !ER             ;RSER SHOULD = 157017
1370  003746  112777  000001  175140        MOVB    #1,@RSER        ;A MOVB INST
1371  003754  022777  000001  175132        CMP     #1,@RSER        ;SHOULD MODIFY COMPLETE WD
1372  003762  001401                        BEQ     .+4             ;OK
1373  003764  104002                        HLT     !ER
1374
1375  003766  104400                TST20:  SCOPE
1376
1377  003770  012777  052005  175116        MOV     #52005,@RSER    ;SET THESE BITS
1378  003776  022777  052005  175110        CMP     #52005,@RSER    ;DID THEY SET
1379  004004  001401                        BEQ     .+4             ;YES
1380  004006  104002                        HLT     !ER             ;ER SHOULD = 52005
1381  004010  104400                TST21:  SCOPE
1382
1383  004012  012777  125012  175074        MOV     #125012,@RSER   ;SET THESE BITS
1384  004020  022777  125012  175066        CMP     #125012,@RSER   ;DID THEY SET
1385  004026  001401                        BEQ     .+4             ;YES
1386  004030  104002                        HLT     !ER             ;ER SHOULD = 105012
```

```
1387  004032  104400                    TST22:  SCOPE
1388
1389  004034  012777  177017  175052            MOV     #177017,@RSER   ;SET THESE BITS
1390  004042  005077  175046                    CLR     @RSER           ;CLEAR THEM
1391  004046  005777  175042                    TST     @RSER           ;DID THEY CLEAR
1392  004052  001401                            BEQ     .+4             ;YES
1393  004054  104002                            HLT     !ER             ;SHOULD = 0
1394  004056  104400                    TST23:  SCOPE
1395
1396                                    ;SET AND CLEAR RSMR
1397
1398  004060  012777  000070  175036            MOV     #70,@RSMR       ;SET THESE BITS
1399  004066  017737  175032  001214            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
1400  004074  042737  177700  001214            BIC     #177700,WORK    ;CLEAR JUNK
1401  004102  022737  000070  001214            CMP     #70,WORK        ;DID THEY SET
1402  004110  001401                            BEQ     .+4             ;YES
1403  004112  104220                            HLT     !MR             ;SHOULD = 70
1404  004114  104400                    TST24:  SCOPE
1405
1406  004116  012777  000070  175000            MOV     #70,@RSMR       ;SET BITS
1407  004124  005077  174774                    CLR     @RSMR           ;CLEAR THEM
1408  004130  032777  000077  174766            BIT     #77,@RSMR       ;DID THEY CLEAR
1409  004136  001401                            BEQ     .+4             ;YES
1410  004140  104220                            HLT     !MR             ;BITS (77) SHOULD = 0
1411  004142  104400                    TST25:  SCOPE
1412
1413  004144  012777  000050  174752            MOV     #50,@RSMR       ;SET BITS
1414  004152  017737  174746  001214            MOV     @RSMR,WORK      ;PUT IN WORKABLE REG
1415  004160  042737  177700  001214            BIC     #177700,WORK    ;CLEAR JUNK
1416  004166  022737  000050  001214            CMP     #50,WORK        ;DID THESE BITS SET
1417  004174  001401                            BEQ     .+4             ;YES
1418  004176  104220                            HLT     !MR             ;BITS (50) SHOULD BE SET
1419  004200  104400                    TST26:  SCOPE
1420
1421  004202  012777  000020  174714            MOV     #20,@RSMR       ;SET BITS
1422  004210  017737  174710  001214            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
1423  004216  042737  177700  001214            BIC     #177700,WORK    ;CLEAR JUNK
1424  004224  022737  000020  001214            CMP     #20,WORK        ;DID THEY SET
1425  004232  001401                            BEQ     .+4             ;YES
1426  004234  104220                            HLT     !MR             ;MR SHOULD AT LEAST HAVE A (21)
```

```
1427                                    ;*****************************************************************************
1428                                    ;TEST 27                TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
1429                                    ;*****************************************************************************
1430  004236  104400                    TST27:  SCOPE
1431
1432  004240  104414                    BITST:  CLRDK                           ;CLEAR ALL RS REG
1433  004242  012777  003566  174630            MOV     #3566,@RSCS1            ;LOAD CSI
1434  004250  112777  000005  174656            MOVB    #5,@RSCS1B              ;LOAD BIT
1435  004256  022777  006766  174614            CMP     #6766,@RSCS1            ;DID IT LOAD?
1436  004264  001401                           BEQ     .+4                     ;YES
1437  004266  104001                           HLT     !CS1
1438  004270  112777  000032  174602            MOVB    #32,@RSCS1
1439  004276  022777  006632  174574            CMP     #6632,@RSCS1
1440  004304  001401                           BEQ     .+4
1441  004306  104001                           HLT     !CS1                    ;CS1 SHOULD = 6632
1442
1443  004310  104400                    TST30:  SCOPE
1444
1445  004312  013777  001160  174562    BITCS2: MOV     UNNUM,@RSCS2            ;LOAD UNIT NUMBER
1446  004320  052777  177400  174554            BIS     #177400,@RSCS2          ;LOAD ALL BITS
1447  004326  105077  174604                    CLRB    @RSCS2B                 ;CLR UPPER BYTE
1448  004332  013701  001160                    MOV     UNNUM,GOOD              ;GET UNIT NO.
1449  004336  052701  000100                    BIS     #100,GOOD               ;SET OR BIT
1450  004342  017700  174534                    MOV     @RSCS2,BAD              ;GET CS2
1451  004346  020001                           CMP     BAD,GOOD                ;IS CS2 CORRECT?
1452  004350  001401                           BEQ     .+4                     ;YES
1453  004352  104000                           HLT                             ;LOAD BYTE DID NOT WORK
1454
1455  004354  104400                    TST31:  SCOPE
1456
1457  004356  012777  025252  174520    BITWC:  MOV     #25252,@RSWC            ;LOAD WC
1458  004364  112777  000377  174546            MOVB    #377,@RSWCB             ;LOAD BIT
1459  004372  022777  177652  174504            CMP     #177652,@RSWC           ;DID IT LOAD?
1460  004400  001401                           BEQ     .+4                     ;YES
1461  004402  104010                           HLT     !WC                     ;NO WC SHOULD =177652
1462  004404  112777  000123  174472            MOVB    #123,@RSWC
1463  004412  022777  177523  174464            CMP     #177523,@RSWC
1464  004420  001401                           BEQ     .+4
1465  004422  104010                           HLT     !WC                     ;WC SHOULD = 177523
1466
1467  004424  104400                    TST32:  SCOPE
1468
1469  004426  012777  025252  174452    BITBA:  MOV     #25252,@RSBA            ;LOAB DA
1470  004434  112777  000377  174500            MOVB    #377,@RSBAB             ;LOAD BIT
1471  004442  022777  177652  174436            CMP     #177652,@RSBA           ;DID IT LOAD?
1472  004450  001401                           BEQ     .+4                     ;YES
1473  004452  104020                           HLT     !BA                     ;DA SHOULD =177652
1474  004454  112777  000125  174424            MOVB    #125,@RSBA
1475  004462  022777  177524  174416            CMP     #177524,@RSBA
1476  004470  001401                           BEQ     .+4
1477  004472  104020                           HLT     !BA                     ;BA SHOULD = 177525
1478  004474  104414                           CLRDK                           ;CLEAR ALL RS REG
```

```
1479                                    ;*****************************************************************************
1480                                    ;TEST 33                TEST DATA LATE IN CS2
1481                                    ;*****************************************************************************
1482  004476  104400                    TST33:  SCOPE
1483
1484                                    ;DO A READ FROM SILO: SHOULD GET DLT + TRE ERROR BECAUSE SILO IS EMPTY
1485
1486  004500  104414                    SILOB:  CLRDK                   ;CLEAR ALL RS REG
1487  004502  017700  174414                    MOV     @RSDB,BAD       ;READ FROM EMPTY SILO
1488  004506  017700  174370                    MOV     @RSCS2,BAD      ;GET CS2
1489  004512  012701  100100                    MOV     #100100,GOOD    ;GET CORRECT ANS
1490  004516  053701  001160                    BIS     UNNUM,GOOD      ;FOR CS2
1491  004522  020001                            CMP     BAD,GOOD        ;IS CS2 CORRECT?
1492  004524  001401                            BEQ     .+4             ;YES
1493  004526  104200                            HLT     !CS2            ;SHOULD HAVE DLT ERROR
1494  004530  022777  144200  174342            CMP     #144200,@RSCS1  ;DID SC AND TRE SET?
1495  004536  001401                            BEQ     .+4             ;YES
1496  004540  104001                            HLT     !CS1            ;SC AND TRE SHOULD BE SET
1497  004542  012777  040000  174330            MOV     #TRE,@RSCS1     ;CLEARS ERROR BIT
1498  004550  032777  140000  174322            BIT     #140000,@RSCS1  ;DID SC + TRE CLEAR
1499  004556  001401                            BEQ     .+4             ;YES
1500  004560  104001                            HLT     !CS1            ;TRE AND SC SHOULD BE 0
1501  004562  017700  174314                    MOV     @RSCS2,BAD      ;GET CS2
1502  004566  042701  100000                    BIC     #BIT15,GOOD     ;GET CORRECT ANS
1503  004572  020100                            CMP     GOOD,BAD        ;IS CS2 CORRECT?
1504  004574  001401                            BEQ     .+4             ;YES
1505  004576  104200                            HLT     !CS2            ;DLT SHOULD BE 0
1506                                    ;*****************************************************************************
1507                                    ;TEST 34                LOAD RSDB WITH ALL ONES AND ALL ZEROS
1508                                    ;*****************************************************************************
1509  004600  104400                    TST34:  SCOPE
1510
1511  004602  104414                    ZERONE: CLRDK                   ;CLEAR ALL RS REG
1512  004604  005077  174312                    CLR     @RSDB           ;LOAD DB WITH ALL 0
1513  004610  012777  177777  174304            MOV     #177777,@RSDB   ;LOAD DB WITH ALL ONES
1514  004616  012737  002000  001214            MOV     #2000,WORK      ;TIME OUT ROUTINE
1515  004624  012701  000300                    MOV     #300,GOOD       ;GET CORRECT FOR CS2
1516  004630  053701  001160                    BIS     UNNUM,GOOD
1517  004634  017700  174242            2$:     MOV     @RSCS2,BAD      ;GET CS2
1518  004640  020100                            CMP     GOOD,BAD        ;IS IT CORRECT?
1519  004642  001404                            BEQ     3$              ;YES
1520  004644  005337  001214                    DEC     WORK            ;TO WAIT FOR OR
1521  004650  001371                            BNE     2$              ;TO SET
1522  004652  104200                            HLT     !CS2            ;OR SHOULD BE SET
1523  004654  005001                    3$:     CLR     GOOD
1524  004656  017700  174240                    MOV     @RSDB,BAD       ;LOAD BAD WITH DB
1525  004662  020100                            CMP     GOOD,BAD        ;IS BAD CORRECT
1526  004664  001401                            BEQ     .+4             ;YES
1527  004666  104000                            HLT                     ;COULD NOT FLOAT 0 THROUGH DB
1528  004670  012701  177777                    MOV     #-1,GOOD        ;LOAD GOOD WITH ANS
1529  004674  017700  174222                    MOV     @RSDB,BAD       ;GET DATA FROM DB
1530  004700  020100                            CMP     GOOD,BAD        ;IS DB CORRECT
1531  004702  001401                            BEQ     .+4             ;YES
1532  004704  104000                            HLT                     ;BAD SHOULD = 1777777
```

```
1533  004706  104400                    TST35:  SCOPE
1534                                    ;TEST FOR 66 LOCATIONS IN SILO PUT COUNT IN EVERY LOCATION
1535
1536  004710  104414                    SILO:   CLRDK                   ;CLEAR ALL RS REG
1537  004712  005001                            CLR     R1              ;CLEAR COUNTER
1538  004714  005201                    1$:     INC     R1              ;INCREMENT COUNTER
1539  004716  010177  174200                    MOV     R1,@RSDB        ;LOAD SILO
1540  004722  022701  000102                    CMP     #66.,R1         ;LAST LOC. YET?
1541  004726  001372                            BNE     1$              ;NO LOOP AGAIN
1542  004730  012701  000200                    MOV     #200,GOOD       ;GET CORRECT ANS FOR CS2
1543  004734  053701  001160                    BIS     UNNUM,GOOD
1544  004740  017700  174136                    MOV     @RSCS2,BAD      ;GET CS2
1545  004744  020100                            CMP     GOOD,BAD        ;IS CS2 CORRECT?
1546  004746  001401                            BEQ     .+4             ;YES
1547  004750  104200                            HLT     !CS2            ;OR SHOULD BE 1
1548  004752  005001                            CLR     GOOD            ;CLEAR LOCATION COUNTER
1549  004754  005201                    2$:     INC     GOOD            ;ADD 1 TO IT
1550  004756  022701  000103                    CMP     #67.,GOOD       ;LAST LOC YET?
1551  004762  001405                            BEQ     3$              ;YES
1552  004764  017700  174132                    MOV     @RSDB,BAD       ;GET LOC FROM DB
1553  004770  020100                            CMP     GOOD,BAD        ;DO LOCATIONS MATCH?
1554  004772  001770                            BEQ     2$              ;YES
1555  004774  104000                            HLT                     ;CAN NOT MATCH 66 LOCATIONS
1556  004776  032777  000200  174076  3$:      BIT     #OR,@RSCS2      ;IS OR 0
1557  005004  001401                            BEQ     .+4             ;YES
1558  005006  104200                            HLT     !CS2            ;OR SHOULD BE 0
1559
1560                                    ;NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT ERROR
1561
1562  005010  005001                            CLR     R1              ;CLEAR COUNTER
1563  005012  005201                    4$:     INC     R1              ;ADD 1 TO COUNT
1564  005014  010177  174102                    MOV     R1,@RSDB        ;PUT INTO COUNTER
1565  005020  022701  000103                    CMP     #67.,R1         ;DONE YET?
1566  005024  001401                            BEQ     .+4             ;YES
1567  005026  000771                            BR      4$              ;NO DO AGAIN
1568  005030  032777  100000  174044            BIT     #DLT,@RSCS2     ;DID DATA LATE SET?
1569  005036  001001                            BNE     .+4             ;YES
1570  005040  104200                            HLT     !CS2            ;DLT DID NOT SET
1571
1572                                    ;DOES SILO CHANGE WITH 67TH WORD: IT SHOULD NOT
1573
1574  005042  017700  174054                    MOV     @RSDB,BAD       ;GET 1ST WD FORM SILO
1575  005046  012701  000001                    MOV     #1,GOOD         ;CORRECT ANS OF SILO
1576  005052  020100                            CMP     GOOD,BAD        ;IS SILO GOOD
1577  005054  001401                            BEQ     .+4             ;YES
1578  005056  104000                            HLT                     ;SILO SHOULD NOT HAVE MOVED
1579  005060  104400                    TST36:  SCOPE
```

```
1580                                    ;FLOAT A 1 AND A 0 THROUGH THE SILO
1581
1582  005062  104414                    SILOFL: CLRDK                   ;CLEAR ALL RS REG
1583  005064  000241                            CLC                     ;CLEAR CARRY TO FLOAT A 0
1584  005066  012701  000001                   MOV     #1,GOOD          ;GET UP DATA FOR INPUT TO SILO
1585  005072  010177  174024            1$:     MOV     GOOD,@RSDB       ;LOAD DB
1586  005076  006101                            ROL     GOOD             ;SHIFT BIT
1587  005100  103401                            BCS     .+4              ;DONE YET SHIFTING?
1588  005102  000773                            BR      1$               ;NO
1589  005104  012701  177776                   MOV     #-2,GOOD         ;SET ALL ONES
1590  005110  000261                            SEC                      ;SET CARRY TO ROL
1591  005112  010177  174004            3$:     MOV     GOOD,@RSDB       ;LOAD SILO
1592  005116  006101                            ROL     GOOD             ;SHILT 0
1593  005120  103774                            BCS     3$               ;LOOP TILL DONE
1594
1595                                    ;NOW TEST OUTPUT
1596  005122  000241                            CLC                      ;CLEAR CARRY
1597  005124  012701  000001                   MOV     #1,GOOD          ;CORRECT ANS
1598  005130  017700  173766            2$:     MOV     @RSDB,BAD        ;GET DATA FROM DB
1599  005134  020100                            CMP     GOOD,BAD         ;IS DB DATA GOOD?
1600  005136  001401                            BEQ     .+4              ;YES
1601  005140  104000                            HLT                      ;DB COULD NOT BUBBLE CORRECTLY
1602  005142  006101                            ROL     GOOD             ;SETUP FOR NEXT ANS
1603  005144  103401                            BCS     .+4              ;DONE YET?
1604  005146  000770                            BR      2$               ;NO
1605  005150  012701  177776                   MOV     #-2,GOOD         ;SETUP FOR ANS
1606  005154  017700  173742            4$:     MOV     @RSDB,BAD        ;GET DATA FROM DB
1607  005160  020100                            CMP     GOOD,BAD         ;IS IT CORRECT?
1608  005162  001401                            BEQ     .+4              ;YES
1609  005164  104000                            HLT                      ;DB WRONG
1610  005166  000261                            SEC                      ;SET CARRY TO ROL
1611  005170  006101                            ROL     GOOD             ;SETUP FOR NEXT ANS
1612  005172  103770                            BCS     4$               ;LOOP TILL DONE
1613                                  ;TEST INTERRUPT IN THE RH11
1614                                  ;BY MOVING 300 INTO RHCS1
1615                                  ;******************************************************************************
1616                                  ;TEST 37                TEST INTERRUPT IN RH11
1617                                  ;******************************************************************************
1618  005174  104400                    TST37:  SCOPE
1619  005176  104414                    INT:    CLRDK                   ;CLEAR ALL ERRORS
1620  005200  012777  005252  173722            MOV     #PGTRAP,@RSVEC   ;SET UP VECTOR
1621  005206  012777  000340  173716            MOV     #340,@RSVCPS     ;SET TRAP PS
1622  005214  012737  000200  177776            MOV     #200,@#PS        ;SET PS AT PRIORITY 4
1623  005222  012777  000300  173650            MOV     #300,@RSCS1      ;THIS SHOULD CAUSE A TRAP
1624  005230  012737  000500  001214            MOV     #500,WORK        ;SETUP LOOP
1625  005236  005337  001214            1$:     DEC     WORK             ;DEC LOOP SHOULD
1626  005242  001375                            BNE     1$               ;INTERRUPT BEFORE LOOP IS DONE
1627  005244  104001                            HLT     !CS1             ;SHOULD NEVER GET HERE
1628  005246  000137  005266                   JMP     INTDON           ;GET OUT
1629  005252  022626                    PGTRAP: CMP     (6)+,(6)+        ;TRAP OK
1630  005254  022777  004200  173616            CMP     #4200,@RSCS1     ;DID IE CLEAR?
1631  005262  001401                            BEQ     .+4              ;YES
1632  005264  104001                            HLT     !CS1             ;IE SHOULD BE CLEARED
1633  005266                            INTDON:
```

```
                                ;*****************************************************************************
                                ;TEST 40                    MAINTENANCE TIMING TEST
                                ;*****************************************************************************
005266  104400                  TST40:  SCOPE

                                ;MODULE TESTED G092
                                ;THE FOLLOWING TEST ON THE RS04 DISK IS A SINGLE-STEPPED
                                ;MAINTENANCE MODE TEST ON THE RS04 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
005270  104414                  MRTIME: CLRDK                   ;CLEAR DRIVE REGISTERS
005272  052737  001040  001166          BIS     #1040,ONCEE     ;SET CLK CNT
005300  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
005302  104420                          MRCK                    ;CHECK MAINTENANCE REG FOR
005304  022701                          22701                   ;22701
005306  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
005310  104430                          MRIND                   ;SEND MAINT INDEX PULSE

005312  104420                          MRCK                    ;CHECK MAINT REG TO
005314  022701                          22701                   ;EQUAL 22701
005316  104000                          HLT                     ;MR=BAD GOOD=CORRECTIONS
                                                                ;COULD NOT INITIALIZE MR REG
                                ;INDEX PULSE SHOULD CLEAR LOOK-AHEAD REG

005320  005777  173574                  TST     @RSLA           ;IS RSLA CLEARED
005324  001401                          BEQ     .+4             ;YES
005326  104224                          HLT     !MR!LA          ;RSLA SHOULD BE CLEARED
                                                                ;WITH THE INDEX PULSE

                                ;PERFORM MAINTENANCE CLOCK OPERATION 1024 TIMES TO
                                ;PROVIDE CLOCK TO STEP TIMING THRU RESYNC PERIOD.
                                ;IF SECTOR PULSE IS ASSERTED DURING THIS LOOP
                                ;CHECK SECTOR BOUNDARY COUNTER AND E12

005330  012737  001000  001202          MOV     #512.,REPT
005336  104422                  MRTIM1: MRCLK                   ;CLOCK MAINT REG WITH AN 11 AND A 1
005340  104420                          MRCK                    ;CHECK MR REG TO
005342  072701                          72701                   ;EQUAL 72701
005344  104000                          HLT                     ;MR = BAD, GOOD = CORRECT ANS
005346  104422                          MRCLK                   ;CLOCK MR
005350  104420                          MRCK                    ;CHECK MR TO
005352  022701                          22701                   ;EQUAL 22701
005354  104000                          HLT                     ;BAD=MR REG GOOD=CORRECTIONS
005356  005337  001202                  DEC     REPT            ;IS THE LOOP DONE YET?
005362  001365                          BNE     MRTIM1          ;NO-LOOP
```

```
                                        ;AFTER ONE MORE CLOCK SECTOR PULSE SHOULD BE ASSERTED
                                        ;IF NOT, CHECK SECTOR BOUNDARY COUNTER, SECTOR BOUNDARY FF (E21) AND E12

005364  104422                                MRCLK                   ;CLOCK MAINT REG WITH A 11 AND A 1
005366  104420                                MRCK                    ;CHECK MR REG TO
005370  072301                                72301                   ;EQUAL 72301
005372  104000                                HLT                     ;MR=BAD GOOD=CORRECTIONS
005374  104422                                MRCLK                   ;CLOCK MR WITH 11 AND A 1
005376  104420                                MRCK                    ;CHECK MAINT REG
005400  022301                                22301                   ;TO EQUAL 22301
005402  104000                                HLT                     ;MR=BAD GOOD-CORRECT ANS
005404  005777  173510                        TST     @RSLA           ;DOES LOOK AHEAD REG=0
005410  001401                                BEQ     MRT2            ;YES-CONT
005412  104224                                HLT     !MR!LA          ;LOOK AHEAD REG SHOULD=0
                                        ;PERFORM MAINTENANCE CLOCK OPERATION 80 TIMES TO PROVIDE
                                        ;CLOCK PULSES TO STEP THRU 1ST SECTOR PRE-AMBLE AREA

005414  005002                        MRT2:   CLR     R2              ;CLEAR R2 FOR SECTOR COMPARE WITH LA REG
005416  012737  000050  001202                MOV     #40.,REPT       ;80 CLOCKS TO STEP THRU PRE-AMBLE
005424  104422                        MRT2A:  MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005426  104420                                MRCK                    ;CHECK MAINT REG
005430  073701                                73701                   ;EQUAL 73701
005432  104000                                HLT                     ;MR = BAD GOOD = CORRECT ANS
005434  104422                                MRCLK                   ;CLOCK MR REG
005436  104420                                MRCK                    ;CHECK MR REG
005440  023701                                23701                   ;TO EQUAL 23701
005442  104000                                HLT                     ;MR = BAD GOOD = CORRECTANS
005444  005337  001202                        DEC REPT                ;REPEAT
005450  001365                                BNE     MRT2A           ;LOOP 40 TIMES

                                        ;SUPPLY CLOCKS TO STEP THROUGH THE DATA AREA IN THE SECTOR
005452  012737  002200  001202                MOV     #9.*128.,REPT   ;18 CLOCKS PER DATA WORD
005460  104422                        MRT2B:  MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005462  104420                                MRCK                    ;CHECK MAINT REG
005464  073701                                73701                   ;TO EQUAL 73701
005466  104000                                HLT                     ;MR = BAD GOOD = CORRECT ANS
005470  104422                                MRCLK                   ;CLOCK MR REG
005472  104420                                MRCK                    ;CHECK MR REG
005474  023701                                23701                   ;TO EQUAL 23701
005476  104000                                HLT                     ;MR=BAD GOOD=CORRECTANS
005500  005337  001202                        DEC     REPT            ;REPEAT
005504  001365                                BNE     MRT2B           ;LOOP
```

```
                                        ;SUPPLY ENOUGH MAINT CLOCKS TO STEP THROUGH THE CRC AREA
                                        ;AND THE DEAD BAND ON THE SECTOR
                                        
005506  012737  000214  001202          MOV     #140.,REPT      ;AMOUNT OF CLOCKS TO END OF SECTOR
005514  104422                  MRT2C:  MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005516  104420                          MRCK                    ;CHECK MAINT REG
005520  073701                          73701                   ;TO EQUAL 73701
005522  104000                          HLT                     ;MR = BAD GOOD = CORRECT ANS
005524  104422                          MRCLK                   ;CLOCK MR REG
005526  104420                          MRCK                    ;CHECK MAINT REG
005530  023701                          23701                   ;TO EQUAL 23701
005532  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
005534  005337  001202                  DEC     REPT            ;REPEAT
005540  001365                          BNE     MRT2C           ;LOOP
005542  104422                          MRCLK                   ;CLOCK MR REG
005544  104420                          MRCK                    ;CHECK MR REG
005546  073701                          73701                   ;TO EQUAL 73701
005550  104000                          HLT                     ;MR = BAD GOOD = CORRECT ANS
                                        ;ONE MORE CLOCK SHOULD CAUSE SECTOR PULSE
                                        ;IF NOT, CHECK E16-6
                                        
005552  104422                          MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005554  104420                          MRCK                    ;MAINT REG SHOULD
005556  023701                          23701                   ;EQUAL 22301
005560  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
005562  104422                          MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005564  104420                          MRCK                    ;MAINT REG
005566  072301                          72301                   ;SHOULD EQUAL 72301
005570  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
                                        
                                        ;LOOK-AHEAD REGISTER SHOULD NOW POINT TO SECTOR 1 (OR 4000 IF INTERLEAVED)
                                        
005572  022777  000002  173326          CMP     #2,@RSDT        ;INTERLEAVED?
005600  001403                          BEQ     3$              ;NO
005602  062702  004000                  ADD     #4000,R2        ;YES
005606  000402                          BR      2$              ;CONT
005610  062702  000100          3$:     ADD     #100,R2         ;INCREMENT SECTOR COMPARE
005614  020277  173300          2$:     CMP     R2,@RSLA        ;LA REG SHOULD=100
005620  001401                          BEQ     1$              ;LA IS CORRECT
005622  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.

005624  012737  000076  001204  1$:     MOV     #62.,REPT1
005632  012737  005152  001202  MRT3:   MOV     #2666.,REPT
005640  104422                  3$:     MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005642  005337  001202                  DEC     REPT            ;STEP THROUGH
005646  001374                          BNE     3$              ;SECTOR
005650  104422                          MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005652  104420                          MRCK                    ;MAINT REG
005654  022701                          22701                   ;SHOULD EQUAL 22701
005656  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
005660  104422                          MRCLK                   ;1 MORE CLK, SAME SECTOR PULSE
005662  104420                          MRCK                    ;MAINT REG SHOULD
005664  072301                          72301                   ;EQUAL 72301
005666  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
005670  022777  000002  173230          CMP     #2,@RSDT        ;DRIVE INTERLEAVED?
005676  001420                          BEQ     6$              ;NO
005700  032737  001000  001166          BIT     #BIT9,ONCEE     ;DO I ADD 4000
005706  001406                          BEQ     4$              ;OR SUBTRACT IT FROM WHAT I EXPECT TO
005710  042737  001000  001166          BIC     #BIT9,ONCEE     ;FIND IN RSLA
005716  162702  004000                  SUB     #4000,R2
005722  000406                          BR      6$
005724  052737  001000  001166  4$:     BIS     #BIT9,ONCEE
005732  062702  004000                  ADD     #4000,R2
005736  000402                          BR      5$
005740  062702  000100          6$:     ADD     #100,R2         ;INCREMENT SECTOR COMPARE
005744  017700  173150          5$:     MOV     @RSLA,BAD       ;LA REG SHOULD HAVE INCREMENTED TO NEXT SECTOR
005750  010201                          MOV     R2,GOOD         ;GET CORRECT ANS FOR RSLA
005752  020100                          CMP     GOOD,BAD        ;COMPARE FOR CORRECT ANS
005754  001401                          BEQ     1$              ;RSLA IS GOOD
005756  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS

005760  005337  001204          1$:     DEC     REPT1           ;REPEAT 62
005764  001322                          BNE     MRT3            ;TIMES
005766  012737  005152  001202          MOV     #2666.,REPT     ;COUNT FOR LAST SECTOR
005774  104422                  2$:     MRCLK                   ;CLOCK
005776  005337  001202                  DEC     REPT            ;THRU
006002  001374                          BNE     2$              ;LAST SECTOR
006004  017700  173110                  MOV     @RSLA,BAD       ;GET CONTENTS OF RSLA
006010  012701  007777                  MOV     #7777,GOOD      ;GET CORRECT ANS
006014  020100                          CMP     GOOD,BAD        ;DOES RSLA EQUAL 7777
006016  001401                          BEQ     .+4             ;YES
006020  104000                          HLT                     ;BAD=RSLA GOOD=CORRECT ANS
```

```
                                ;*****************************************************************************
                                ;TEST 41                    SECTOR FRACTION TEST
                                ;*****************************************************************************
006022  104400                  TST41:  SCOPE
                                ;MODULE TESTED G092
                                ;CLOCK THROUGH AN ENTIRE TRACK IN MAINT MODE WHILE
                                ;CHECKING FOR THE PROPER OPERATION OF THE LOOK-AHEAD REGISTER AND
                                ;THE SECTOR FRACTION COUNTER. 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.

006024  104414                  MRT4:   CLRDK                   ;CLEAR DRIVE REGISTERS
006026  052737  000040  001166          BIS     #40,ONCEE       ;SET FLAG BITS
006034  042737  003000  001166          BIC     #3000,ONCEE
006042  005037  001174                  CLR     MCCNT           ;CLEAR MAINT CLOCK COUNTER
006046  005002                          CLR     R2              ;CLEAR R2 FOR SECTOR COUNTER
006050  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
006052  104420                          MRCK                    ;CHECK MR REG
006054  022701                          22701                   ;TO EQUAL 22701
006056  104424                          MRINT                   ;INIT MAINT MODE
006060  104430                          MRIND                   ;ISSUE A MAINT INDEX PULSE
                                                                ;TO CLEAR THE DRIVE
006062  104420                          MRCK                    ;CHECK MAINT REG
006064  022701                          22701                   ;TO EQUAL 22701
006066  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;ISSUE 1024 MAINT CLOCKS TO STEP THROUGH THE RESYNC AREA

006070  012737  001000  001202          MOV     #512.,REPT      ;COUNT TO STEP THRU RESYNC AREA
006076  104422  MRT4A:                 MRCLK                   ;CLOCK THROUGH RESYNC
006100  104420                          MRCK                    ;CHECK MAINT REG
006102  072701                          72701                   ;TO EQUAL 72701
006104  104000                          HLT                     ;MR = BAD GOOD = CORRECT ANS
006106  005777  173006                  TST     @RSLA           ;IS RSLA=TO 0
006112  001401                          BEQ     .+4             ;YES
006114  104204                          HLT     !LA             ;RSLA SHOULD=0 DURING RESPONSE
006116  104422                          MRCLK                   ;CLOCK MR REG
006120  104420                          MRCK                    ;CHECK MR REG
006122  022701                          22701                   ;TO EQUAL 22701
006124  104000                          HLT                     ;BAD=MR GOOD=CORRECT ANS
006126  005777  172766                  TST     @RSLA           ;IS RSLA=TO 0
006132  001401                          BEQ     .+4             ;YES
006134  104204                          HLT     !LA             ;RSLA SHOULD=0 DURING RESPONSE
006136  005337  001202                  DEC     REPT            ;LOOP THROUGH
006142  001355                          BNE     MRT4A           ;RESYNC AREA

                                ;ONE MORE PULSE SHOULD CAUSE THE FIRST SECTOR PULSE

006144  104422                          MRCLK                   ;CLOCK MR WITH A 11 AND A 1
006146  104420                          MRCK                    ;CHECK MAINT REG FOR SECTOR PULSE
006150  072301                          72301                   ;MR SHOULD=72301
006152  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
1866  006154  104422                  MRT4B:  MRCLK                   ;CLOCK MR REG WITH A 11 AND A 1
1867  006156  104420                          MRCK                    ;CHECK MAINT REG
1868  006160  022301                          22301                   ;TO EQUAL 22301
1869  006162  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
1870
1871                                  ;SECTOR FRACTION BITS IN LOOK-AHEAD REGISTER SHOULD BE CLEARED (EQUAL TO 00)
1872
1873  006164  017700  172730                  MOV     @RSLA,BAD       ;GET RSLA
1874  006170  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1875  006172  020100                          CMP     GOOD,BAD        ;IS THE RSLA REG CORRECT
1876  006174  001401                          BEQ     1$              ;YES
1877  006176  104000                          HLT                     ;RSLA=BAD GOOD=CORRECTANS
1878
1879                                  ;STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA
1880                                  ;AREA WHILE CHECKING THE SECTOR FRACTION
1881
1882  006200  012737  000244  001202  1$:     MOV     #164.,REPT      ;FOR FIRST FRACTION CHANGE
1883  006206  104422                  MRT4C:  MRCLK                   ;CLOCK MR REG WITH A 11 AND A 1
1884  006210  017700  172704                  MOV     @RSLA,BAD       ;GET RSLA
1885  006214  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1886  006216  020001                          CMP     BAD,GOOD        ;IS RSLA CORRECT
1887  006220  001401                          BEQ     1$              ;YES
1888  006222  104000                          HLT                     ;BAD=RSLA GOOD=CORRECT ANS
1889  006224  005337  001202          1$:     DEC     REPT            ;LOOP ON
1890  006230  001366                          BNE     MRT4C           ;PREAMBLE AREA
1891
1892                                  ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1893
1894  006232  104422                          MRCLK                   ;CLOCK MR WITH A 11 AND A 1
1895  006234  005202                          INC     R2              ;COUNT THE FRACTION
1896  006236  017700  172656                  MOV     @RSLA,BAD       ;GET RSLA
1897  006242  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1898  006244  020001                          CMP     BAD,GOOD        ;IS RSLA CORRECT?
1899  006246  001401                          BEQ     2$              ;YES
1900  006250  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS
1901
1902                                  ;FIRST FRACTION CHANGES AFTER 164 MAINT. CLKS, THE REST
1903                                  ;CHANGE AFTER 40 MAINTENANCE CLOCKS
1904
1905  006252  012737  000076  001202  2$:     MOV     #62.,REPT       ;COUNT FOR WORDS IN A SECTOR
1906  006260  012737  000047  001204  MRT4D:  MOV     #39.,REPT1      ;COUNT FOR SECT FRACT TO CHANGE
1907  006266  104422                  MRT4E:  MRCLK                   ;CLOCK MR WITH A 11 AND A 1
1908  006270  017700  172624                  MOV     @RSLA,BAD       ;GET RSLA
1909  006274  010201                          MOV     R2,GOOD         ;GET CORRECT ANS
1910  006276  020100                          CMP     GOOD,BAD        ;IS RSLA CORRECT?
1911  006300  001401                          BEQ     1$              ;YES
1912  006302  104000                          HLT                     ;RSLA=BAD GOOD=CORRECT ANS
1913  006304  005337  001204          1$:     DEC     REPT1           ;LOOP
1914  006310  001366                          BNE     MRT4E
```

```
1915                                        ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1916
1917  006312  104422                        MRCLK                   ;CLOCK MR WITH A 11 AND A 1
1918  006314  022702  007777                CMP     #7777,R2        ;AT THE LAST SECTOR-LAST FRACTION?
1919  006320  001472                        BEQ     MRT4F           ;YES, FINISH THE SECTOR
1920  006322  005202                        INC     R2              ;NO, ADD 1 TO FRACTION
1921  006324  017700  172570          4$:   MOV     @RSLA,BAD       ;GET RSLA
1922  006330  022777  000002  172570        CMP     #2,@RSDT        ;IS THIS DIRVE INTERLEAVED?
1923  006336  001431                        BEQ     12$             ;NO
1924  006340  032737  002000  001166        BIT     #BIT10,ONCEE    ;HAS REPT GONE TO ZERO YET FOR THIS SECTOR?
1925  006346  001425                        BEQ     12$             ;NO
1926                                        ;RSLA NOW POINTS TO NEXT INTERLEAVED SECTOR BIT 9 IN ONCEE
1927                                        ;INDICATES WHETHER RSLA SHOULD NOW BE BETWEEN
1928                                        ;0000-3700(1) OR 4000-7700(0).
1929  006350  032737  001000  001166        BIT     #BIT9,ONCEE     ;SHOULD RSLA BE BETWEEN 0-3700?
1930  006356  001004                        BNE     9$              ;YES
1931  006360  052737  001000  001166        BIS     #BIT9,ONCEE     ;SET FOR NEXT PASS
1932  006366  000406                        BR      10$
1933  006370  042737  001000  001166  9$:   BIC     #BIT9,ONCEE     ;CLEAR FOR NEXT PASS
1934  006376  042702  004000                BIC     #4000,R2        ;MAKE EXPECTED RSLA LESS THAN 4000
1935  006402  000404                        BR      5$
1936  006404  062702  004000          10$:  ADD     #4000,R2        ;COMPENSATE FOR INTERLEAVING
1937  006410  162702  000100                SUB     #100,R2
1938  006414  042737  002000  001166  5$:   BIC     #BIT10,ONCEE    ;CLEAR FLAG FOR NEXT SECTOR
1939  006422  010201                  12$:  MOV     R2,GOOD         ;GET CORRECT ANSWER FOR RSLA
1940  006424  020100                        CMP     GOOD,BAD        ;IS RSLA CORRECT
1941  006426  001401                        BEQ     2$              ;YES
1942  006430  104000                        HLT                     ;RSLA=BAD GOOD=CORRECT ANS
1943  006432  005337  001202          2$:   DEC     REPT            ;HAS SECTOR FRACTION REACHED 77?
1944  006436  001310                        BNE     MRT4D           ;NO
1945
1946                                        ;CHECK FOR END OF ONE SECTOR OR BEGINNING OF NEXT
1947
1948  006440  010203                  11$:  MOV     R2,R3
1949  006442  042703  177700                BIC     #177700,R3      ;CHECK SECTOR FRACTION
1950  006446  022703  000077                CMP     #77,R3          ;END OF SECTOR?
1951  006452  001402                        BEQ     3$              ;YES
1952  006454  000137  006154                JMP     MRT4B           ;NO, BEGINNING OF NEXT
1953  006460  012737  000025  001204  3$:   MOV     #21.,REPT1      ;SETUP LOOP TO FINISH
1954  006466  012737  000001  001202        MOV     #1,REPT         ;THIS SECTOR
1955  006474  052737  002000  001166        BIS     #BIT10,ONCEE    ;REPT HAS GONE TO ZERO FOR THIS SECTOR
1956  006502  000137  006266                JMP     MRT4E           ;LOOP
1957
1958  006506  012737  000021  001202  MRT4F:MOV     #17.,REPT
1959  006514  104422                  1$:   MRCLK                   ;CLOCK MR WITH A 11 AND A 1
1960  006516  017700  172376                MOV     @RSLA,BAD       ;GET RSLA
1961  006522  010201                        MOV     R2,GOOD         ;R2 SHOULD=7777
1962  006524  020100                        CMP     GOOD,BAD        ;IS RSLA CORRECT-END OF DISK?
1963  006526  001401                        BEQ     2$              ;YES
1964  006530  104000                        HLT                     ;RSLA=BAD GOOD=CORRECT ANS (7777)
1965  006532  005337  001202          2$:   DEC     REPT            ;FINISH
1966  006536  001366                        BNE     1$              ;LOOP
```

```
                                        ;SECTOR AND FRACTION IS = TO 7777 TO INDICATE LAST WORD ON THIS TRACK
                                        ;RSLA SHOULD EQUAL 7700 ON ANOTHER MAINT CLOCK.
                                        
006540  104422          MRT4G:  MRCLK                   ;CLOCK MR WITH A 11 AND A 1
006542  017700  172352          MOV     @RSLA,BAD       ;GET RSLA
006546  012701  007700          MOV     #7700,GOOD      ;GET CORRECT ANS
006552  020100                  CMP     GOOD,BAD        ;IS RSLA CORRECT?
006554  001401                  BEQ     1$              ;YES
006556  104000                  HLT                     ;RSLA=BAD GOOD=CORRECT ANS
006560  104430          1$:     MRIND                   ;ISSUE AN INDEX PULSE TO
                                                        ;CLEAR THE DRIVE
006562  017700  172332          MOV     @RSLA,BAD       ;GET RSLA
006566  005001                  CLR     GOOD            ;GET CORRECT ANS
006570  020100                  CMP     GOOD,BAD        ;IS RSLA CORRECT?
006572  001401                  BEQ     2$              ;YES
006574  104000                  HLT                     ;RSLA=BAD GOOD=CORRECT ANS
006576  104420          2$:     MRCK                    ;CHECK MR REG
006600  022701                  22701                   ;TO EQUAL 22701
006602  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                ;*****************************************************************************
                                ;TEST 42                    ILLEGAL FUNCTION TEST
                                ;*****************************************************************************
006604  104400                  TST42:  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

006606  104414                  MRILF:  CLRDK                   ;CLEAR ALL THE DRIVE REGISTERS
006610  042737  000040  001166          BIC     #BIT5,ONCEE     ;CLEAR CLOCK CNT FLAG
006616  032737  000002  001166          BIT     #BIT1,ONCEE     ;WAS THERE AN ERROR
006624  001002                          BNE     MRLF1           ;YES DO NOT CHANGE "ILF" CODE
006626  012702  000003                  MOV     #3,R2           ;SETUP FIRST "ILF" CODE
                                ;PUT DRIVE IN MAINTENANCE MODE

006632  104416                  MRLF1:  MRDMD                   ;PUT DRIVE INTO MAINT MODE
006634  104420                          MRCK                    ;CHECK MR REG TO
006636  022701                          22701                   ;EQUAL 22701
006640  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)

                                ;ASSERT A MAINTENANCE MODE DISK "INDEX" PULSE

006642  104430                          MRIND
006644  010277  172230          MRLF2:  MOV     R2,@RSCS1       ;SEND "ILF" WITH THE "GO" BIT
006650  017700  172236                  MOV     @RSDS,BAD       ;GET DRIVE STATUS REG
006654  012701  150600                  MOV     #150600,GOOD    ;GET CORRECT ANS
006660  020100                          CMP     GOOD,BAD        ;IS RSDS CORRECT?
006662  001440                          BEQ     1$              ;YES
006664  104402  006670                  TYPE    ,+2             ;.ASCIZ <15><12>"ILLEGAL FUNCTION CODE SENT TO DRIVE= "
006740  010237  001214                  MOV     R2,WORK         ;GET FUNCTION CODE
006744  013746  001214                  MOV     WORK,-(6)       ;PUT WORK ON STACK
006750  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
006752  052737  000002  001166          BIS     #BIT1,ONCEE     ;SET ERROR BIT SO ILLEGAL FUN DOESN'T CHANGE
006760  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
006762  104040                          HLT     !DS

006764  042737  000002  001166  1$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
006772  017700  172116                  MOV     @RSER,BAD       ;GET RSER
006776  012701  000001                  MOV     #1,GOOD         ;GET CORRECT ANS
007002  020100                          CMP     GOOD,BAD        ;DID "ILF" SET IN RSER
007004  001404                          BEQ     2$              ;YES
007006  052737  000002  001166          BIS     #BIT1,ONCEE     ;SET ERROR BIT
007014  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
007016  042737  000002  001166  2$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
```

```
                                        ;CLEAR THE DRIVE FOR THE NEXT "ILF" CODE PASS
007024  104414                  MRCILF: CLRDK                   ;CLEAR ERRORS
007026  017700  172060                  MOV     @RSDS,BAD       ;GET RSDS REG
007032  012701  010600                  MOV     #10600,GOOD     ;GET CORRECT ANS
007036  020100                          CMP     GOOD,BAD        ;DID "ATA" AND "ERR" CLEAR IN RSDS?
007040  001435                          BEQ     1$              ;YES
007042  104402  007046                  TYPE    ,.+2            ;.ASCIZ <15><12>"ATA AND ERR IN RSDS SHOULD CLEAR WITH I
007124  052737  000002  001166          BIS     #BIT1,ONCEE
007132  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
007134  042737  000002  001166  1$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
007142  017700  171746                  MOV     @RSER,BAD       ;GET RSER
007146  005001                          CLR     GOOD            ;GET CORRECT ANS
007150  020100                          CMP     GOOD,BAD        ;DID ILF CLEAR IN RSER
007152  001431                          BEQ     2$              ;YES
007154  052737  000002  001166          BIS     #BIT1,ONCEE     ;SET ERROR BIT
007162  104402  007166                  TYPE    ,.+2            ;.ASCIZ <15><12>"ILF IN RSER SHOULD CLEAR WITH INIT"
007234  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
007236  042737  000002  001166  2$:     BIC     #BIT1,ONCEE     ;CLEAR ERROR BIT
                                        ;GET NEXT ILLEGAL FUNCTION COE

007244  062702  000002          MRLF3:  ADD     #2,R2           ;UPDATE ILF
007250  022702  000011                  CMP     #11,R2          ;IS THIS A ILF CODE
007254  001773                          BEQ     MRLF3           ;NO-UPDATE IT
007256  022702  000021                  CMP     #21,R2
007262  001770                          BEQ     MRLF3
007264  022702  000031                  CMP     #31,R2
007270  001765                          BEQ     MRLF3
007272  022702  000051                  CMP     #51,R2
007276  001762                          BEQ     MRLF3
007300  022702  000061                  CMP     #61,R2
007304  001757                          BEQ     MRLF3
007306  022702  000071                  CMP     #71,R2
007312  001754                          BEQ     MRLF3
007314  022702  000101                  CMP     #101,R2
007320  001402                          BEQ     ILFDON          ;FINISHED ALL ILF CODES GET OUT
007322  000137  006632                  JMP     MRLF1           ;START NEXT ILF FUNCTION
007326                          ILFDON:
```

```
                                ;*****************************************************************************
                                ;TEST 43                    TEST NO-OP CODES 1 AND 21
                                ;*****************************************************************************
007326  104400                  TST43:  SCOPE

                                ;MODULE TESTED M7759
007330  104414                  MROP:   CLRDK                   ;CLEAR ALL DRIVE REGISTERS
007332  042737  000004  001166          BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
007340  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
007342  104420                          MRCK                    ;CHECK MR REG TO
007344  022701                          22701                   ;EQUAL 22701
007346  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;SEND INDEX PULSE
007350  032737  000010  001166          BIT     #BIT3,ONCEE     ;TESTING CODE I
007356  001031                          BNE     3$              ;NO CODE 21
007360  012777  000001  171512          MOV     #1,@RSCS1       ;LOAD NO-OP FUNCTION
007366  012737  000001  001214          MOV     #1,WORK         ;LOAD NO-OP FUNCTION
007374  005777  171514                  TST     @RSER           ;ANY ERRORS
007400  001403                          BEQ     1$              ;NO
007402  004737  021724                  JSR     PC,NOPERR       ;TYPE IT
007406  104040                          HLT     !D$             ;TYPE ERROR
007410  022777  010600  171474  1$:     CMP     #10600,@RSDS    ;IS RSDS CORRECT
007416  001403                          BEQ     2$              ;YES
007420  004737  021724                  JSR     PC,NOPERR       ;RSDS SHOULD
007424  104040                          HLT     !D$             ;EQUAL 10600
007426  042737  000004  001166  2$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG

                                ;TEST NO-OP FUNCTION CODE 21

007434  052737  000010  001166          BIS     #BIT3,ONCEE     ;TEST TESTING CODE 21 FLAG
007442  012737  000021  001214  3$:     MOV     #21,WORK        ;LOAD CODE 21
007450  012777  000021  171422          MOV     #21,@RSCS1      ;LOAD FUNCTION
007456  005777  171432                  TST     @RSER           ;ANY ERRORS?
007462  001403                          BEQ     4$              ;NO
007464  004737  021724                  JSR     PC,NOPERR       ;YES, TYPE ERROR
007470  104040                          HLT     !D$             ;ERROR DURING NO-OP FUNCTION
007472  022777  010600  171412  4$:     CMP     #10600,@RSDS    ;IS RSDS CORRECT
007500  001403                          BEQ     5$              ;YES
007502  004737  021724                  JSR     PC,NOPERR       ;TYPE ERROR
007506  104040                          HLT     !D$             ;RSDS SHOULD=10600
007510  042737  000014  001166  5$:     BIC     #14,ONCEE       ;CLEAR TEST BITS
```

```
                                ;*****************************************************************************
                                ;TEST 44                    TEST NO-OP FUNCTION WITH ERROR BITS SET
                                ;*****************************************************************************
007516  104400                  TST44:  SCOPE

                                ;MODULE TESTED M7759
007520  104414                  MROPER: CLRDK                   ;CLEAR ALL REGISTERS
007522  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
007524  104420                          MRCK                    ;CHECK MR REG
007526  022701                          22701                   ;TO EQUAL 22701
007530  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
007532  104430                          MRIND                   ;SEND INDEX PULSE

007534  012777  177777  171352          MOV     #-1,@RSER       ;LOAD RSER WITH ERRORS
007542  013701  001164                  MOV     UNCMP,GOOD      ;GET DRIVE UNDER TEST
007546  042701  177400                  BIC     #177400,GOOD
007552  017700  171340                  MOV     @RSAS,BAD       ;GET RSAS REG
007556  020100                          CMP     GOOD,BAD        ;DID ATA BIT SET CAUSED BY ERROR
007560  001427                          BEQ     1$              ;YES
007562  104402  007566                  TYPE    ,.+2            ;.ASCIZ <15><12>"SET ERRORS IN RSER-RSAS IS INCORRECT"
007636  104000                          HLT                     ;RSAS=BAD GOOD=CORRECT ANS
007640  012737  000001  001214  1$:     MOV     #1,WORK         ;SETUP FOR NO-OP CODE 1
007646  032737  000010  001166          BIT     #BIT3,ONCEE     ;TESTING CODE 21?
007654  001004                          BNE     2$              ;YES
007656  012777  000001  171214          MOV     #1,@RSCS1       ;SEND NO-OP CODE 1
007664  000406                          BR      3$              ;CHECK FOR ERRORS
007666  012737  000021  001214  2$:     MOV     #21,WORK        ;SETUP FOR CODE 21
007674  012777  000021  171176          MOV     #21,@RSCS1      ;SENT NO-OP CODE 21
007702  017700  171206          3$:     MOV     @RSER,BAD       ;GET RSER REG
007706  012701  177017                  MOV     #177017,GOOD    ;GET CORRECT ANS
007712  020100                          CMP     GOOD,BAD        ;DID RSER CHANGE WITH NO-OP
007714  001411                          BEQ     4$              ;NO
007716  104402  007722                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSER "
007732  004737  022020                  JSR     PC,CHG
007736  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
007740  017700  171152          4$:     MOV     @RSAS,BAD       ;GET RSAS
007744  013701  001164                  MOV     UNCMP,GOOD      ;GET CORRECT ANS
007750  042701  177400                  BIC     #177400,GOOD
007754  020100                          CMP     GOOD,BAD        ;IS RSAS CORRECT
007756  001411                          BEQ     5$              ;YES
007760  104402  007764                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSAS "
007774  004737  022020                  JSR     PC,CHG          ;TYPE ERROR
010000  104000                          HLT                     ;RSAS=BAD GOOD=CORRECT ANS
010002  017700  171104          5$:     MOV     @RSDS,BAD       ;GET RSDS
010006  012701  150600                  MOV     #150600,GOOD    ;GET CORRECT ANS
010012  020100                          CMP     GOOD,BAD        ;DID RSDS CHANGE
010014  001411                          BEQ     6$              ;NO
010016  104402  010022                  TYPE    ,.+2            ;.ASCIZ <15><12>"RSDS "
010032  004737  022020                  JSR     PC,CHG          ;TYPE ERROR
010036  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
010040  032737  000010  001166  6$:     BIT     #BIT3,ONCEE     ;TESTING CODE 21
010046  001005                          BNE     7$              ;YES, GET OUT
010050  052737  000010  001166          BIS     #BIT3,ONCEE     ;SET CODE 21 FLAG
010056  000137  007520                  JMP     MROPER          ;TEST CODE 21
010062  042737  000010  001166  7$:     BIC     #BIT3,ONCEE     ;DONE CLEAR FLAG AND CONT.
```

```
                                ;*****************************************************************************
                                ;TEST 45                    BLOCK SEARCH TEST 1
                                ;*****************************************************************************
010070  104400                  TST45:  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.

010072  104414          MRSRCH: CLRDK                   ;CLEAR ALL REGISTERS
010074  052737  000040  001166          BIS     #BIT5,ONCEE     ;SET CLOCK FLAG
010102  104416                        MRDMD                   ;PUT DRIVE INTO MAINTENANCE MOE
010104  104420                        MRCK                    ;CHECK MR REG
010106  022701                        22701                   ;TO EQUAL 22701
010110  104424                        MRINT                   ;INIT MR REG (CLEAR MRSP)
010112  104430                        MRIND                   ;CLOCK INDEX PULSE IN RSMR
010114  012777  000003  170766        MOV     #3,@RSDA        ;DO A SEARCH FOR SECTOR 3 OR 41
010122  022777  000002  170776        CMP     #2,@RSDT        ;INTERLEAVED?
010130  001403                        BEQ     4$              ;NO SECTOR 3
010132  012777  000041  170750        MOV     #41,@RSDA       ;YES SECTOR 41
010140  012777  000031  170732  4$:   MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759)
010146  104426                        DSCK                    ;CHECK RSDS
010150  030400                        30400                   ;TO EQUAL 30400
010152  104000                        HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                                ;BE 0 FOR A DRIVE SEARCH CMD
010154  012737  021506  001202        MOV     #21506,REPT     ;STEP THROUGH 3 SECTORS
010162  104422          1$:   MRCLK                   ;CLOCK MR
010164  104426                        DSCK                    ;RSDS SHOULD NOT
010166  030400                        30400                   ;CHANGE TILL CLOCKING IS COMPLETED
010170  104000                        HLT                     ;TO REACH SECTOR 3
010172  005337  001202                DEC     REPT            ;KEEP CLOCKING TILL
010176  001371                        BNE     1$              ;SECTOR 3 HAS BEEN REACHED
                                ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
010200  104422                        MRCLK                   ;CLOCK MR REG
010202  104426                        DSCK                    ;CHECK FOR "ATA" AND "DRY"
010204  110600                        110600                  ;TO BE SET IN RSDS FOR
010206  104000                        HLT                     ;SEARCH FUNCTION SHOULD BE COMPLETED
010210  022777  104230  170662        CMP     #104230,@RSCS1  ;SET RSCS1
010216  001401                        BEQ     2$              ;SC IN RSCS1 SHOULD SET BECAUSE OF
010220  104140                        HLT     !DS!AS          ;COMPLETED SEARCH FUNCTION
010222  013777  001162  170666  2$:   MOV     UNITSV,@RSAS    ;CLEAR ATA
010230  005777  170662                TST     @RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
010234  001401                        BEQ     3$              ;YES
010236  104140                        HLT     !DS!AS          ;RSAS SHOULD=0
010240  022777  004230  170632  3$:   CMP     #4230,@RSCS1    ;DID SC CLEAR BY CLEARING
010246  001401                        BEQ     .+4             ;"ATA" YES
010250  104140                        HLT     !DS!AS          ;NO
```

```
                                ;*************************************************************************
                                ;TEST 46                    BLOCK SEARCH TEST 2
                                ;*************************************************************************
010252  104400                  TST46:  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.

010254  104414                  MRSRC:  CLRDK                   ;CLEAR ALL REGISTERS
010256  052737  000040  001166          BIS     #BIT5,ONCEE     ;SET CLOCK FLAG
010264  104416                          MRDMD                   ;PUT DRIVE INTO MAINTENANCE MOE
010266  104420                          MRCK                    ;CHECK MR REG
010270  022701                          22701                   ;TO EQUAL 22701
010272  104424                          MRINT                   ;INIT MR REG (CLEAR MRSP)
                                ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
010274  104430                          MRIND
010276  104420                          MRCK                    ;CHECK MR REG TO EQUAL
010300  022701                          22701                   ;22701
010302  104000                          HLT                     ;
                                ;STEP THRU RESYNC PERIOD
010304  012737  001000  001202          MOV     #512.,REPT
010312  052737  000040  001166          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
010320  104422                  MRRT1:  MRCLK                   ;CLOCK MR REG
010322  104420                          MRCK                    ;CHECK FOR
010324  072701                          72701                   ;CORRECT DATA
010326  104000                          HLT                     ;MR = BAD GOOD = CORRECT DATA
010330  104422                          MRCLK                   ;CLOCK MR REG
010332  104420                          MRCK                    ;CHECK FOR
010334  022701                          22701                   ;CORRECT DATA
010336  104000                          HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
010340  005337  001202                  DEC     REPT            ;FINISH LOOPING
010344  001365                          BNE     MRRT1           ;THROUGH RESYNC PERIOD
                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE SP = 0
010346  104422                          MRCLK                   ;CLOCK MR REG
010350  104420                          MRCK                    ;MR SHOULD
010352  072301                          72301                   ;EQUALS 72301
010354  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
010356  104422                          MRCLK                   ;CLOCK MR REG
010360  104420                          MRCK                    ;CHECK MR
010362  022301                          22301                   ;TO EQUAL 22301
010364  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
010366  012737  000100  001202          MOV     #100,REPT       ;STEP INTO SECTOR 0
010374  104422                  2$:     MRCLK                   ;CLOCK MR REG
010376  005337  001202                  DEC     REPT            ;DO 100 TIMES
010402  001374                          BNE     2$              ;DONE YET? NO BR
010404  012777  000031  170466  4$:     MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759) FOR SECTOR 0
010412  104426                          DSCK                    ;CHECK RSDS
010414  030400                          30400                   ;TO EQUAL 30400
010416  104000                          HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                                ;BE 0 FOR A DRIVE SEARCH CMD
010420  012737  021506  001202          MOV     #21506,REPT     ;STEP 3 SECTORS BEYOND SECTOR 0
```

```
2274  010426  104422                    1$:     MRCLK                   ;CLOCK MR
2275  010430  104426                            DSCK                    ;RSDS SHOULD NOT
2276  010432  030400                            30400                   ;CHANGE TILL CLOCKING IS COMPLETED
2277  010434  104000                            HLT                     ;TO REACH SECTOR 3
2278  010436  005337  001202                    DEC     REPT            ;KEEP CLOCKING TILL
2279  010442  001371                            BNE     1$              ;SECTOR 3 HAS BEEN REACHED
2280                                    ;ASSERT INDEX PULSE TO SIMULATE THE BEGINNING OF THE NEXT REVOLUTION
2281  010444  104430                            MRIND
2282  010446  104420                            MRCK                    ;CHECK MR REG TO EQUAL
2283  010450  022701                            22701                   ;22701
2284  010452  104000                            HLT
2285
2286                                    ;STEP THRU RESYNC PERIOD
2287
2288  010454  012737  001000  001202            MOV     #512.,REPT
2289  010462  052737  000040  001166            BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2290  010470  104422                    MRWR1:  MRCLK                   ;CLOCK MR REG
2291  010472  104420                            MRCK                    ;CHECK FOR
2292  010474  072701                            72701                   ;CORRECT DATA
2293  010476  104000                            HLT                     ;MR = BAD GOOD = CORRECT DATA
2294  010500  104422                            MRCLK                   ;CLOCK MR REG
2295  010502  104420                            MRCK                    ;CHECK FOR
2296  010504  022701                            22701                   ;CORRECT DATA
2297  010506  104000                            HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2298  010510  005337  001202                    DEC     REPT            ;FINISH LOOPING
2299  010514  001365                            BNE     MRWR1           ;THROUGH RESYNC PERIOD
2300
2301                                    ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2302                                    ;SP=0 EQUALS SECTOR PULSE
2303  010516  104422                            MRCLK                   ;CLOCK MR REG
2304  010520  104420                            MRCK                    ;MR SHOULD
2305  010522  072301                            72301                   ;EQUAL 72301
2306  010524  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
2307  010526  104422                            MRCLK                   ;CLOCK MR REG
2308  010530  104420                            MRCK                    ;CHECK MR
2309  010532  022301                            22301                   ;TO EQUAL 22301
2310  010534  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
2311
2312                                    ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2313  010536  104422                            MRCLK                   ;CLOCK MR REG
2314  010540  104426                            DSCK                    ;CHECK FOR "ATA" AND "DRY"
2315  010542  110600                            110600                  ;TO BE SET IN RSDS FOR
2316  010544  104000                            HLT                     ;SEARCH FUNCTION SHOULD BE COMPLETED
2317  010546  022777  104230  170324            CMP     #104230,@RSCS1  ;SET RSCS1
2318  010554  001401                            BEQ     2$              ;SC IN RSCS1 SHOULD SET BECAUSE OF
2319  010556  104140                            HLT     !DS!AS          ;COMPLETED SEARCH FUNCTION
2320  010560  013777  001162  170330    2$:     MOV     UNITSV,@RSAS    ;CLEAR ATA
2321  010566  005777  170324                    TST     @RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
2322  010572  001401                            BEQ     3$              ;YES
2323  010574  104140                            HLT     !DS!AS          ;RSAS SHOULD=0
2324  010576  022777  004230  170274    3$:     CMP     #4230,@RSCS1    ;DID SC CLEAR BY CLEARING
2325  010604  001401                            BEQ     .+4             ;"ATA" YES
2326  010606  104140                            HLT     !DS!AS  ;NO
```

```
2327                                    ;*************************************************************************
2328                                    ;TEST 47                DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)
2329                                    ;*************************************************************************
2330  010610  104400                    TST47:  SCOPE
2331
2332                                    ;MODULE TESTED M7759, M7755, M7770
2333                                    ;RMR ERROR IS CAUSED BY WRITTING INTO RSCS1 WHILE DOING A BLOCK SEARCH FUNCTION
2334                                    ;CHECK RMR DECODER, E12, M7755, IF THIS TEST FAILS
2335
2336  010612  104414                    RMRC1:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
2337  010614  042737  000040  001166            BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
2338  010622  104416                            MRDMD                   ;PUT DRIVE INTO MAINT MODE
2339  010624  104420                            MRCK                    ;CHECK MR REG TO
2340  010626  022701                            22701                   ;EQUAL 22701
2341  010630  104424                            MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
2342  010632  012777  000001  170250            MOV     #1,@RSDA        ;LOAD RSDA
2343  010640  012777  000031  170232            MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
2344  010646  104426                            DSCK                    ;CHECK RSDS
2345  010650  030400                            30400                   ;TO EQUAL 30400
2346  010652  104000                            HLT                     ;DRY IN RSDS SHOULD BE
2347                                                                    ;CLEARED FOR DRIVE WAS
2348                                                                    ;ISSURED A BLOCK SEARCH FUNCTION
2349                                                                    ;RSDS=BAD GOOD=CORRECT ANS
2350  010654  012777  000011  170216            MOV     #11,@RSCS1      ;LOAD A CLEAR FUNCTION
2351                                                                    ;THIS SHOULD CAUSE AN RMR
2352                                                                    ;ERROR FOR DRIVE WAS BUSY
2353                                                                    ;WHEN CLEAR COMMAND WAS GIVEN
2354  010662  017700  170226                    MOV     @RSER,BAD       ;GET RSER REG
2355  010666  012701  000004                    MOV     #4,GOOD         ;GET CORRECT ANS
2356  010672  020100                            CMP     GOOD,BAD        ;DID RMR SET IN RSER?
2357  010674  001410                            BEQ     1$              ;YES
2358  010676  104402  022071                    TYPE    ,TRMR
2359  010702  104402  010706                    TYPE    ,.+2            ;.ASCIZ "RSCS1"
2360  010714  104000                            HLT                     ;RSER=BAD GOOD=CORRECT ANS
2361  010716  104426                    1$:     DSCK                    ;CHECK RSDS TO
2362  010720  150600                            150600                  ;EQUAL 150600
2363  010722  104000                            HLT                     ;RSDS=BAD GOOD=CORRECT ANS
2364  010724  022777  104230  170146            CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
2365  010732  001401                            BEQ     2$              ;YES
2366  010734  104040                            HLT     !DS             ;RSCS1 SHOULD=104230
2367                                                                    ;RSDS SHOULD=150600
2368                                                                    ;RSER SHOULD=4
2369  010736  022777  000001  170144    2$:     CMP     #1,@RSDA        ;DID CLR CLEAR RSDA
2370  010744  001401                            BEQ     4$              ;NO
2371  010746  104004                            HLT     !DA             ;RSDA SHOULD=1
2372  010750  104414                    4$:     CLRDK                   ;CLEAR ALL REGISTERS
2373  010752  005777  170136                    TST     @RSER           ;RSER SHOULD CLEAR
2374  010756  001401                            BEQ     3$              ;RSER OK
2375  010760  104040                            HLT     !DS             ;RSER SHOULD=0 FOR THE
2376                                                                    ;CLEAR BIT WAS LOADED IN RSCS2
2377  010762  022777  004200  170110    3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
2378  010770  001401                            BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
2379  010772  104040                            HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*****************************************************************************
                                ;TEST 50                    DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)
                                ;*****************************************************************************
010774  104400                TST50:  SCOPE

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

010776  104414                RMRC2:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
011000  104416                        MRDMD                   ;PUT DRIVE INTO MAINT MODE
011002  104420                        MRCK                    ;CHECK MR REG TO
011004  022701                        22701                   ;EQUAL 22701
011006  104424                        MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
011010  012777  000001  170072        MOV     #1,@RSDA        ;LOAD RSDA
011016  012777  000031  170054        MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
011024  104426                        DSCK                    ;CHECK RSDS
011026  030400                        30400                   ;TO EQUAL 30400
011030  104000                        HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
011032  005077  170052                CLR     @RSDA           ;MODIFY RSDA
                                                                ;THIS SHOULD CAUSE AN RMR
                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                ;WHEN COMMAND WAS GIVEN
011036  017700  170052                MOV     @RSER,BAD       ;GET RSER REG
011042  012701  000004                MOV     #4,GOOD         ;GET CORRECT ANS
011046  020100                        CMP     GOOD,BAD        ;DID RMR SET IN RSER?
011050  001410                        BEQ     1$              ;YES
011052  104402  022071                TYPE    ,TRMR
011056  104402  011062                TYPE    ,.+2            ;.ASCIZ "RSDA"
011070  104000                        HLT                     ;RSER=BAD GOOD=CORRECT ANS
011072  104426                1$:     DSCK                    ;CHECK RSDS TO
011074  150600                        150600                  ;EQUAL 150600
011076  104000                        HLT                     ;RSDS=BAD GOOD=CORRECT ANS
011100  022777  104230  167772        CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
011106  001401                        BEQ     2$              ;YES
011110  104040                        HLT     !DS             ;RSCS1 SHOULD=104230
                                                                ;RSDS SHOULD=50400
                                                                ;RSER SHOULD=4
011112  022777  000001  167770  2$:   CMP     #1,@RSDA        ;DID CLR CLEAR RSDA
011120  001401                        BEQ     4$              ;NO
011122  104004                        HLT     !DA             ;RSDA SHOULD=1
011124  104414                4$:     CLRDK                   ;CLEAR ALL REGISTERS
011126  005777  167762                TST     @RSER           ;RSER SHOULD CLEAR
011132  001401                        BEQ     3$              ;RSER OK
011134  104040                        HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
011136  022777  004200  167734  3$:   CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
011144  001401                        BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
011146  104040                        HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*************************************************************************
                                ;TEST 51                 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)
                                ;*************************************************************************
011150  104400                  TST51:  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.

011152  104414                  RMRC3:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
011154  042737  000040  001166          BIC     #BIT5,ONCEE     ;CLEAR CLOCK COUNT FLAG
011162  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
011164  104420                          MRCK                    ;CHECK MR REG TO
011166  022701                          22701                   ;EQUAL 22701
011170  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
011172  012777  000001  167710          MOV     #1,@RSDA        ;LOAD RSDA
011200  012777  000031  167672          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
011206  104426                          DSCK                    ;CHECK RSDS
011210  030400                          30400                   ;TO EQUAL 30400
011212  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
011214  012777  177777  167672          MOV     #-1,@RSER       ;MODIFY RSER
                                                                ;THIS SHOULD CAUSE AN RMR
                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                ;WHEN COMMAND WAS GIVEN
011222  017700  167666                  MOV     @RSER,BAD       ;GET RSER REG
011226  012701  000004                  MOV     #4,GOOD         ;GET CORRECT ANS
011232  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
011234  001410                          BEQ     1$              ;YES
011236  104402  022071                  TYPE    ,TRMR
011242  104402  011246                  TYPE    ,.+2            ;.ASCIZ "RSER"
011254  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
011256  104426                  1$:     DSCK                    ;CHECK RSDS TO
011260  150600                          150600                  ;EQUAL 150600
011262  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
011264  022777  104230  167606          CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
011272  001401                          BEQ     4$              ;YES
011274  104040                          HLT     !DS             ;RSCS1 SHOULD=104230
                                                                ;RSDS SHOULD=150600
                                                                ;RSER SHOULD=4
011276  104414                  4$:     CLRDK                   ;CLEAR ALL REGISTERS
011300  005777  167610                  TST     @RSER           ;RSER SHOULD CLEAR
011304  001401                          BEQ     3$              ;RSER OK
011306  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LORDED IN RSCS2
011310  022777  004200  167562  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
011316  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
011320  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*****************************************************************************
                                ;TEST 52                    DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)
                                ;*****************************************************************************
011322  104400                  TST52:  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.

011324  104414                  RMRC4:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
011326  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
011330  104420                          MRCK                    ;CHECK MR REG TO
011332  022701                          22701                   ;EQUAL 22701
011334  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
011336  012777  000001  167544          MOV     #1,@RSDA        ;LOAD RSDA
011344  012777  000031  167526          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
011352  104426                          DSCK                    ;CHECK RSDS
011354  030400                          30400                   ;TO EQUAL 30400
011356  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
011360  005077  167532                  CLR     @RSAS           ;WRITE INTO ATTENTION SUMMARY REGISTER.
                                                                ;SHOULD BE NO RMR ERROR BECAUSE
                                                                ;WRITING RSAS IS ALLOWED ANYTIME.
011364  017700  167524                  MOV     @RSER,BAD       ;GET RSER REG
011370  012701  000000                  MOV     #0,GOOD         ;GET CORRECT ANS
011374  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
011376  001435                          BEQ     1$              ;NO
011400  104402  011404                  TYPE    ,.+2            ;.ASCIZ <15><12>"RMR ERROR SHOULD NOT SET WHILE WRITING
011470  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
011472  104426                  1$:     DSCK                    ;CHECK RSDS TO
011474  030400                          30400                   ;EQUAL 30400
011476  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
011500  022777  004231  167372          CMP     #4231,@RSCS1    ;DID CORRECT BITS SET IN RSCS1
011506  001401                          BEQ     4$              ;YES
011510  104040                          HLT     !DS             ;RSCS1 SHOULD=4231
                                                                ;RSDS SHOULD=30400
                                                                ;RSER SHOULD=0
011512  104414                  4$:     CLRDK                   ;CLEAR ALL REGISTERS
011514  005777  167374                  TST     @RSER           ;RSER SHOULD CLEAR
011520  001401                          BEQ     3$              ;RSER OK
011522  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
011524  022777  004200  167346  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
011532  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
011534  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                ;*****************************************************************************
                                ;TEST 53                    DRIVE SELECT TEST
                                ;*****************************************************************************
011536  104400                  TST53:  SCOPE

                                ;MODULE TESTED: M7755
                                ;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 ALL 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.
                                ;CHECK UNIT NO. COMPARATOR, E19-M7755 IF TEST FAILS

011540  104414          MRDSEL: CLRDK                   ;CLEAR ALL REGISTERS
011542  104416                  MRDMD                   ;PUT DRIVE INTO MAINT MODE
011544  104420                  MRCK                    ;CHECK MAINT REG
011546  022701                  22701                   ;TO EQUAL 22701
011550  104424                  MRINT                   ;INITIALIZE MAINT MODE (CLEAR MRSP)
                                                        ;BY SENDING 2 CLOCK PULSES
011552  012777  177777  167330  MOV     #-1,@RSDA       ;LOAD DISK ADDR REG OF DRIVE UNDER TEST
                        ;SEARCH FOR NON EXISTENT DRIVES
011560  012737  000401  001214  MOV     #401,WORK
011566  005001                  CLR     GOOD
011570  010177  167306  1$:     MOV     GOOD,@RSCS2     ;LOAD UNIT NO
011574  005777  167314          TST     @RSER           ;IS THIS A NED?
011600  032777  010000  167274  BIT     #BIT12,@RSCS2   ;CHECK
011606  001005                  BNE     2$              ;YES
011610  005201                  INC     GOOD            ;UPDATE UNIT NUMBER
011612  006137  001214          ROL     WORK            ;KEEP LOOKING FOR NED
011616  103460                  BCS     NEDDON          ;COULD NOT FIND ANY NON EXISTENT DRIVES
011620  000763                  BR      1$              ;LOOK FOR NED
011622  012777  004000  167250  2$:     MOV     #4000,@RSCS1    ;CLEAR NED
011630  010137  001220          MOV     GOOD,WORK1      ;SAVE NED NUMBER
011634  010177  167242          MOV     GOOD,@RSCS2     ;LOAD UNIT # OF NED INTO RSCS2
011640  005077  167244          CLR     @RSDA           ;WRITE INTO A NON EXISTENT DRIVE REG
                                                        ;THIS SHOULD CAUSE NED TO
                                                        ;SET IN RSCS2
011644  017700  167232          MOV     @RSCS2,BAD      ;GET RSCS2
011650  052701  010100          BIS     #10100,GOOD     ;PUT CORRECT ANS IN GOOD
                                                        ;BY SETTING NED AND IR
011654  020100                  CMP     GOOD,BAD        ;IS RSCS2 CORRECT?
011656  001401                  BEQ     .+4             ;YES
011660  104000                  HLT                     ;RSCS2=BAD GOOD=CORRECT ANS

011662  022777  160200  167210  CMP     #160200,@RSCS1  ;IS CS1 CORRECT
011670  001401                  BEQ     .+4             ;YES
011672  104004                  HLT     !DA             ;TRE SHOULD BE SET IN CS1 BECAUSE
                                                        ;OF NED ERROR IN RSCS2
                                                        ;RSCS1 SHOULD=160200
```

```
2578  011674  005777  167216                    TST     @RSAS           ;DID ANY ATTENTION BITS SET?
2579  011700  001401                            BEQ     .+4             ;NO
2580  011702  104100                            HLT     !AS             ;NO ATTENTION BITS SHOULD BE SET
2581  011704  112777  000100  167222            MOVB    #100,@RSCS1B    ;CLEAR TRE
2582  011712  032777  010000  167162            BIT     #NED,@RSCS2     ;DID NED CLEAR
2583  011720  001401                            BEQ     .+4             ;YES
2584  011722  104040                            HLT     !DS             ;NED DID NOT CLEAR IN RSCS2
2585                                                                    ;BY CLEARING TRE BIT IN RSCS1
2586  011724  013777  001160  167150            MOV     UNNUM,@RSCS2    ;LOAD CORRECT UNIT NUMBER
2587  011732  022777  177777  167150            CMP     #-1,@RSDA       ;DID RSDA GET MODIFIED
2588                                                                    ;WHILE WRITING INTO A NON
2589                                                                    ;EXISTENT DRIVE?
2590  011740  001443                            BEQ     NNDD            ;NO
2591  011742  104004                            HLT     !DA             ;RSDA SHOULD= -1
2592  011744  013700  001220                    MOV     WORK1,BAD       ;IT GOT MODIFIED WHILE WRITING
2593  011750  013701  001160                    MOV     UNNUM,GOOD      ;INTO A NED
2594  011754  104000                            HLT                     ;GOOD=DRIVE UNDER TEST
2595  011756  000434                            BR      NNDD            ;BAD=NON EXISTENT DRIVE THAT WAS
2596                                                                    ;IN RSCS2 WHEN RSDA GOT MODIFIED
2597  011760  032737  010000  001166  NEDDON:   BIT     #BIT12,ONCEE    ;WAS THIS TYPED BEFORE?
2598  011766  001030                            BNE     NNDD            ;YES
2599  011770  104402  011774                    TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
2600  012042  052737  010000  001166            BIS     #BIT12,ONCEE    ;SET TYPED MESSAGE FLAG
2601  012050                            NNDD:
```

```
                                ;*******************************************************************************
                                ;TEST 54                        MAINTENANCE MODE WRITE TEST
                                ;*******************************************************************************
012050  104400                  TST54:  SCOPE

                                ;MODULE TESTED: M7771, M7753, M7751
                                ;THIS IS AN RSD4 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.

012052  012737  000002  001144          MOV     #2,FLAG2        ;SET TEST FLAG
012060  104414          MRWRT:  CLRDK                           ;CLEAR DRIVE REGISTERS
012062  052737  000040  001166          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
012070  042737  000600  001166          BIC     #600,ONCEE      ;CLEAR FLAG BITS
012076  104430                  MRIND                           ;SEND INDEX PULSE TO MR REG
012100  104420                  MRCK                            ;CHECK MR REG
012102  022701                  22701                           ;TO EQUAL 22701
012104  104424                  MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 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 (110 WORDS)

012106  012702  027566                  MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
012112  005022                  CLR     (R2)+                   ;CLEAR 1ST LOCATION
012114  012722  177777          MOV     #-1,(R2)+               ;2ND WORD OF ALL ONES
012120  005003                  CLR     R3                      ;CLEAR WORK LOC TO GENERATE
012122  000261                  SEC                             ;A PATTERN OF FLOATING ONES
012124  006103          1$:     ROL     R3                      ;GET PATTERN
012126  103402                  BCS     2$                      ;DONE GET OUT
012130  010322                  MOV     R3,(R2)+                ;FILL BUFFER
012132  000774                  BR      1$                      ;CONT
012134  012703  000156          2$:     MOV     #110.,R3        ;FILL REMAINING PORTION OF

012140  012704  146314                  MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
012144  010422          3$:     MOV     R4,(R2)+                ;LOAD BUFFER
012146  005303                  DEC     R3                      ;DONE YET?
012150  001375                  BNE     3$                      ;NO

                                ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) TO SECTOR 0
012152  012777  027566  166726          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
012160  012777  177600  166716          MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
012166  012777  000061  166704          MOV     #61,@RSCS1      ;LOAD WRITE COMMAND
012174  104446                  GETSP                           ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
012176  104220                  HLT     !MR                     ;COULD NOT SET SECTOR PULSE (0)
012200  104450                  SPASS                           ;CLOCK MR 2 TIMES SP = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
012202  104430                                  MRINC
012204  104420                                  MRCK                    ;CHECK MR REG TO EQUAL
012206  020501                                  20501                   ;20501 FOR A
012210  104000                                  HLT                     ;WRITE COMD HAS BEEN ISSUED

                                        ;STEP THRU RESYNC PERIOD

012212  012737  001000  001202                  MOV     #512.,REPT
012220  052737  000040  001166                  BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
012226  104422          MRWRT1:                 MRCLK                   ;CLOCK MR REG
012230  104420                                  MRCK                    ;CHECK FOR
012232  070501                                  70501                   ;CORRECT DATA
012234  104000                                  HLT                     ;MR = BAD GOOD = CORRECT DATA
012236  104422                                  MRCLK                   ;CLOCK MR REG
012240  104420                                  MRCK                    ;CHECK FOR
012242  020501                                  20501                   ;CORRECT DATA
012244  104000                                  HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
012246  005337  001202                          DEC     REPT            ;FINISH LOOPING
012252  001365                                  BNE     MRWRT1          ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
012254  104422                                  MRCLK                   ;CLOCK MR REG
012256  104420                                  MRCK                    ;MR SHOULD
012260  070101                                  70101                   ;EQUAL 70101
012262  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
012264  104422                                  MRCLK                   ;CLOCK MR REG
012266  104420                                  MRCK                    ;CHECK MR
012270  020101                                  20101                   ;TO EQUAL 20101
012272  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 63 DOUBLE MAINT CLOCK OPERATIONS--WRITING PREAMBLE

012274  012737  000077  001202                  MOV     #63.,REPT
012302  104422          MRWRT2:                 MRCLK                   ;CLOCK MR REG
012304  104420                                  MRCK                    ;CHECK MR REG
012306  071501                                  71501                   ;TO EQUAL 71501
012310  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
012312  104422                                  MRCLK                   ;CLOCK MR REG
012314  104420                                  MRCK                    ;CHECK MR REG
012316  021501                                  21501                   ;TO EQUAL 21501
012320  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
012322  005337  001202                          DEC     REPT            ;DONE YET
012326  001365                                  BNE     MRWRT2          ;NO LOOP
```

```
                                ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
                                        MRCLK                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        171501                  ;TO EQUAL 171501
                                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                        MRCLK                   ;CLOCK MR REG
                                        MRCK                    ;MR REG SHOULD
                                        25501                   ;EQUAL 25501
                                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                        MRCLK
                                        MRCK
                                        175501
                                        HLT
                                ;PERFORM NEXT STEP 3 TIMES TO FINISH WRITTING PREAMBLE
                                        MOV     #3,REPT
                                MRWRT3: MRCLK                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        25501                   ;TO EQUAL 25501
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS
                                        MRCLK                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        175501                  ;TO EQUAL 175501
                                        HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                        DEC     REPT            ;DONE YES?
                                        BNE     MRWRT3          ;NO LOOP BACK

                                ;MOVE DATA WORD INTO RS04 SHIFT REGISTER (M7753)

                                        MRCLK                   ;CLOCK MR REG
                                        MRCK                    ;CHECK MR REG
                                        27501                   ;TO EQUAL 27501
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS
                                        MRCLK                   ;CLOCK MR REG
                                        MRCK                    ;MR REG SHOULD
                                        123501                  ;EQUAL 123501
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS

                                ;ENCODE SYNC 1 (M7751)

                                        MRCLK                   ;CLOCK MR REG
                                        MRCK                    ;MR REG SHOULD NOW
                                        73501                   ;EQUAL 73501
                                        HLT                     ;MR=BAD GOOD=CORRECT ANS
                                        MOV     #INBUF,R5       ;GET STARTING ADDR FOR DATA BUFFER
                                        MOV     (R5),R4         ;GET DATA
```

```
2747  012452  012737  002167  001214          MOV     #1143.,WORK     ;DOING A 1 SECTOR TRANSFER 127 WORDS
2748                                                                    ;18 BITS PER WORD-CLOCK LOOPS
2749                                                                    ;TAKE CARE OF 2 BITS AT A TIME
2750                                                                    ;127 TIMES 9 EQUALS 1143 LOOPS
2751                                                                    ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
2752  012460  042737  000200  001166          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
2753  012466  052737  000100  001166          BIS     #BIT6,ONCEE     ;SET 1ST TRANSFER WORD FLAG
2754  012474  104432                  1$:     XBIT                    ;GET 2 BITS OF DATA
2755  012476  104434                          CLKD1                   ;SEND FIRST CLOCK PULSE
2756                                                                    ;AND CALCULATE MR REG
2757                                                                    ;FOR CORRECT DATA (MWDT+MWDB)
2758  012500  104000                          HLT                     ;MR REG NOT CORRECT
2759  012502  104436                          CLKD2                   ;SEND 2ND CLOCK PULSE TO
2760                                                                    ;COMPLETE TRANSFER OF 2 BITS
2761                                                                    ;CALCULATE CORRECT ANS FOR
2762                                                                    ;MR REG (MWDT+MWDB)
2763  012504  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
2764  012506  032737  000200  001166          BIT     #BIT7,ONCEE     ;ON LAST WORD YET?
2765  012514  001015                          BNE     2$              ;YES
2766  012516  032737  000400  001166          BIT     #BIT8,ONCEE     ;ON CRC WORD YET?
2767  012524  001040                          BNE     3$              ;YES
2768  012526  005337  001214                  DEC     WORK            ;DONE WITH 127 WORDS?
2769  012532  001360                          BNE     1$              ;NO
2770
2771  012534  052737  000200  001166          BIS     #BIT7,ONCEE     ;SET LAST WORD FLAG
2772  012542  012737  000012  001214          MOV     #10.,WORK       ;SET UP TO TRANSFER LAST WORD
2773  012550  005337  001214          2$:     DEC     WORK            ;DONE YET?
2774  012554  001347                          BNE     1$              ;NO
2775
2776  012556  052737  000400  001166          BIS     #BIT8,ONCEE     ;SET TRANSFERRING CRC WORD
2777  012564  042737  000200  001166          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
2778  012572  004737  024112                  JSR     PC,GENCRC       ;GENERATE CRC WORD
2779                                                                    ;AND LEAVE IN "WORK"
2780  012576  012702  027566                  MOV     #INBUF,R2       ;GO TO END
2781  012602  062702  000400                  ADD     #400,R2         ;OF DATA BUFFER
2782  012606  013712  001214                  MOV     WORK,@R2        ;LOAD CRC WORD
2783  012612  010205                          MOV     R2,R5           ;RESET POINTER FOR
2784  012614  162705  000002                  SUB     #2,R5           ;R5 FOR CRC WD
2785  012620  012737  000012  001214          MOV     #10.,WORK       ;SETUP TO XFER CRC
2786  012626  005337  001214          3$:     DEC     WORK            ;DONE YET
2787  012632  001320                          BNE     1$              ;NO
```

```
2788                                    ;EBL SHOULD NOW ASSERT
2789
2790  012634  104422                            MRCLK                   ;CLOCK MR REG TO STOP THROUGH
2791                                                                    ;THE RSO4 DISK SECTOR DEAD BAND AREA
2792  012636  104420                            MRCK                    ;CHECK MR REG
2793  012640  153501                            153501                  ;TO EQUAL 103501
2794  012642  104000                            HLT                     ;MR REG=BAD GOOD=CORRECT ANS
2795
2796                                    ;LOOP 6 TIMES
2797
2798  012644  012737  000006  001202            MOV     #6,REPT
2799  012652  104422                    4$:     MRCLK                   ;CLOCK MR REG
2800  012654  104420                            MRCK                    ;CHECK MR REG
2801  012656  003501                            3501                    ;TO EQUAL 53501
2802  012660  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
2803  012662  104422                            MRCLK                   ;CLOCK MR REG
2804  012664  104420                            MRCK                    ;CHECK MR REG
2805  012666  153501                            153501                  ;TO EQUAL 153501
2806  012670  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
2807  012672  005337  001202                    DEC     REPT            ;DONE LOOPING YET?
2808  012676  001365                            BNE     4$              ;NO
2809
2810                                    ;FINISH UP
2811
2812  012700  104422                            MRCLK                   ;CLOCK MR REG
2813  012702  104420                            MRCK                    ;CHECK MR REG
2814  012704  003501                            3501                    ;TO EQUAL 3501
2815  012706  104000                            HLT                     ;MR REG=BAD GOOD=CORRECT ANS
2816  012710  104422                            MRCLK                   ;CLOCK MR REG
2817  012712  104420                            MRCK                    ;CHECK MR REG
2818  012714  151501                            151501                  ;TO EQUAL 151501
2819  012716  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
2820
2821                                    ;TRANSFER SHOULD NOW BE COMPLETE
2822
2823  012720  104422                            MRCLK                   ;CLOCK MR REG
2824  012722  104420                            MRCK                    ;CHECK MRE REG
2825  012724  002701                            2701                    ;TO EQUAL 2701
2826  012726  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
2827
2828                                    ;NOW TEST CONTROLLER
2829
2830  012730  005777  166144                    TST     @RSCS1          ;ANY ERRORS?
2831  012734  100001                            BPL     5$              ;NO
2832  012736  104014                            HLT     !DA!WC          ;YES
2833  012740  005777  166140            5$:     TST     @RSWC           ;DID WC GO TO 0
2834  012744  001401                            BEQ     .+4             ;YES
2835  012746  104010                            HLT     !WC             ;WC SHOULD BE = TO 0
2836  012750  022777  000001  166132            CMP     #1,@RSDA        ;DOES RSDA=1
2837  012756  001401                            BEQ     .+4             ;YES
2838  012760  104004                            HLT     !DA             ;RSDA SHOULD=1
2839  012762  032737  000002  001144            BIT     #BIT1,FLAG2     ;IN MAINT VERIFY TEST
2840  012770  001002                            BNE     .+6             ;NO
2841  012772  000137  021354                    JMP     @#MRVR2         ;YES, GO TO VERIFY TEST
```

```
2842                            ;*****************************************************************************
2843                            ;TEST 55                    MAINTENANCE READ TEST
2844                            ;*****************************************************************************
2845  012776  104400            TST55:  SCOPE
2846
2847                            ;MODULE TESTED: M7771, M7753, M7751
2848                            ;THIS IS AN RS04 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR
2849                            ;READ TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE
2850                            ;DISK DECODING LOGIC TO CORE MEMORY. (THE PHASE LOCK LOOP IS
2851                            ;NOT TESTED IN MAINTENANCE MODE.)
2852
2853  013000  104414            MRRD:   CLRDK                   ;CLEAR DRIVE REGISTERS
2854  013002  052737  000040  001166          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
2855  013010  042737  147716  001166          BIC     #147716,ONCEE   ;CLEAR ALL OTHER FLAG BITS
2856  013016  104430            MRIND                   ;SEND INDEX PULSE TO MR REG
2857  013020  104420            MRCK                    ;CHECK MR REG
2858  013022  022701            22701                   ;TO EQUAL 22701
2859  013024  104424            MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
2860                                                    ;BY SENDING 2 CLOCK PULSES
2861
2862  013026  005037  001144            CLR     FLAG2           ;CLEAR FLAG TEST BITS
2863
2864                            ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
2865                            ;DATA BUFFER WORDS ARE  :A WORD OF ALL 0'S
2866                            ;                       A WORD OF ALL 1'S
2867                            ;                       FLOATING 1'S PATTERN (16 WORDS)
2868                            ;                       A PATTERN OF 146314 (110 WORDS)
2869
2870  013032  012702  027566            MOV     #INBUF,R2       ;GET LOCATION OF INBUF
2871  013036  005022                    CLR     (R2)+           ;CLEAR 1ST LOCATION
2872  013040  012722  177777            MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
2873  013044  005003                    CLR     R3              ;CLEAR WORK LOC TO GENERATE
2874  013046  000261                    SEC                     ;A PATTERN OF FLOATING ONES
2875  013050  006103            1$:     ROL     R3              ;GET PATTERN
2876  013052  103402                    BCS     2$              ;DONE GET OUT
2877  013054  010322                    MOV     R3,(R2)+        ;FILL BUFFER
2878  013056  000774                    BR      1$              ;CONT
2879  013060  012703  000156    2$:     MOV     #110.,R3        ;FILL REMAINING PORTION OF
2880  013064  012704  146314            MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
2881  013070  010422            3$:     MOV     R4,(R2)+        ;LOAD BUFFER
2882  013072  005303                    DEC     R3              ;DONE YET
2883  013074  001375                    BNE     3$              ;NO
2884
2885                            ;NOTE:
2886                            ;INBUF CONTAINS THE TABLE OF DATA WHICH IS "READ".
2887                            ;VIA THE MRDB AND MRDT BITS IN RSMR
2888                            ;OUTBUF IS WHERE THE DATA WORDS FROM THE
                                ;MASSBUS ARE STORED
```

```
2889                                        ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
2890
2891  013076  012777  030366  166002        MOV     #OUTBUF,@RSBA   ;LOAD BUSS ADDR REG
2892  013104  012777  177600  165772        MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
2893  013112  012777  000071  165760        MOV     #71,@RSCS1      ;LOAD READ COMMAND
2894  013120  012702  000200                MOV     #200,R2         ;CLEAR THE OUTBUF TABLE
2895  013124  012703  030366                MOV     #OUTBUF,R3      ;SO THAT WHEN THE READ
2896  013130  005023                4$:     CLR     (R3)+           ;IS FINISHED, WE CAN COMPARE
2897  013132  005302                        DEC     R2              ;WHAT WE GOT (OUTBUF) WITH WHAT
2898  013134  001375                        BNE     4$              ;WE EXPECTED (INBUF).
2899  013136  104446                        GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
2900                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
2901                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
2902  013140  104220                        HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
2903  013142  104450                        SPASS                   ;CLOCK MR REG 2 TIMES SP = 1
2904
2905                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2906  013144  104430                        MRIND
2907  013146  104420                        MRCK                    ;CHECK MR REG TO EQUAL
2908  013150  022601                        22601                   ;22601 FOR A
2909  013152  104000                        HLT                     ;READ COMD
2910
2911                                        ;STEP THRU RESYNC PERIOD
2912
2913  013154  012737  001000  001202        MOV     #512.,REPT
2914  013162  052737  000040  001166        BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERRORS OCCUR
2915  013170  104422                MRRD1:  MRCLK                   ;CLOCK MR REG
2916  013172  104420                        MRCK                    ;CHECK FOR
2917  013174  072601                        72601                   ;CORRECT DATA
2918  013176  104000                        HLT                     ;MR=BAD GOOD=CORRECT DATA
2919  013200  104422                        MRCLK                   ;CLOCK MR REG
2920  013202  104420                        MRCK                    ;CHECK FOR
2921  013204  022601                        22601                   ;CORRECT DATA
2922  013206  104000                        HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
2923  013210  005337  001202                DEC     REPT            ;FINISH LOOPING
2924  013214  001365                        BNE     MRRD1           ;THROUGH RESYNC PERIOD
2925
2926                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2927                                        ;SP=0 EQUALS SECTOR PULSE
2928  013216  104422                        MRCLK                   ;CLOCK MR REG
2929  013220  104420                        MRCK                    ;MR SHOULD
2930  013222  072201                        72201                   ;EQUAL 72201
2931  013224  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
2932  013226  104422                        MRCLK                   ;CLOCK MR REG
2933  013230  104420                        MRCK                    ;CHECK MR
2934  013232  022201                        22201                   ;TO EQUAL 22201
2935  013234  104000                        HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--

013236  012737  000107  001202          MOV     #71.,REPT
013244  104422                  MRRD2:  MRCLK                   ;CLOCK MR REG
013246  104420                          MRCK                    ;CHECK MR REG
013250  073601                          73601                   ;TO EQUAL 73601
013252  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
013254  104422                          MRCLK                   ;CLOCK MR REG
013256  104420                          MRCK                    ;CHECK MR REG
013260  023601                          23601                   ;TO EQUAL 23601
013262  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
013264  005337  001202                  DEC     REPT            ;DONE YET
013270  001365                          BNE     MRRD2           ;NO LOOP
013272  104422                          MRCLK                   ;CLOCK MR REG
013274  104420                          MRCK                    ;CHECK MR REG
013276  073601                          73601                   ;TO EQUAL 73601
013300  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;READ SYNC"1"

013302  012777  000055  165614          MOV     #55,@RSMR
013310  012777  000045  165606          MOV     #45,@RSMR
013316  104420                          MRCK                    ;CHECK MR REG
013320  023645                          23645                   ;TO EQUAL
013322  104000                          HLT                     ;CONTENTS OF GOOD
013324  012777  000055  165572          MOV     #55,@RSMR
013332  012777  000045  165564          MOV     #45,@RSMR
013340  104420                          MRCK
013342  173645                          173645
013344  104000                          HLT

                                        ;READ DATA
013346  005037  001224          MRD3:   CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
013352  012705  027566                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
013356  162705  000002                  SUB     #2,R5
013362  012737  000025  001204          MOV     #21.,REPT1      ;SETUP COUNTER FOR 1ST SB BIT
013370  012737  002200  001202          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;128 WORDS-9X128=1152
                                                                ;2 CLOCKS PER 2 BITS OF DATA
013376  104444                  1$:     RBIT                    ;GET 2 DATA BITS
013400  104440                          CLKR1                   ;CLOCK MR
013402  104000                          HLT                     ;MR REG NOT CORRECT
013404  104442                          CLKR2                   ;CLOCK MR REG
013406  104000                          HLT                     ;MR REG NOT CORRECT
013410  005337  001202                  DEC     REPT            ;DONE WITH DATA BUFFER YET?
013414  001370                          BNE     1$              ;NO
```

```
2982  013416  032737  000400  001166  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
2983  013424  001030                          BNE     3$              ;YES
2984  013426  052737  000400  001166          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
2985  013434  013737  001204  001172          MOV     REPT1,SAVEE     ;SAVE REPT1
2986  013442  004737  024112                  JSR     PC,GENCRC       ;GENERATE CRC WORD
2987                                                                  ;AND LEAVE IN LOC "WORK"
2988  013446  012702  027566                  MOV     #INBUF,R2       ;
2989  013452  013737  001172  001204          MOV     SAVEE,REPT1     ;RESTORE REPT1
2990  013460  062702  000400                  ADD     #400,R2         ;STORE CRC WORD AT END OF
2991  013464  013712  001214                  MOV     WORK,@R2        ;INBUF TABLE
2992  013470  010205                          MOV     R2,R5
2993  013472  162705  000002                  SUB     #2,R5
2994  013476  012737  000011  001202          MOV     #9.,REPT        ;SETUP TO TRANSFER 1 WD
2995  013504  000734                          BR      1$              ;TRANSFER CRC WD
2996  013506  104422                  3$:     MRCLK                   ;CLOCK MR REG
2997  013510  104420                          MRCK                    ;CHECK MR REG
2998  013512  003601                          3601                    ;TO EQUAL
2999  013514  104000                          HLT                     ;3601
3000  013516  104422                          MRCLK                   ;CLOCK MR REG
3001  013520  104420                          MRCK                    ;CHECK MR
3002  013522  153601                          153601                  ;TO EQUAL
3003  013524  104000                          HLT                     ;153601
3004  013526  104422                          MRCLK                   ;CLOCK MR REG
3005  013530  104420                          MRCK                    ;CHECK MR
3006  013532  007601                          7601                    ;TO EQUAL
3007  013534  104000                          HLT                     ;7601
3008  013536  104422                          MRCLK                   ;CLOCK MR REG
3009  013540  104420                          MRCK                    ;CHECK MR
3010  013542  153601                          153601                  ;TO EQUAL
3011  013544  104000                          HLT                     ;153601
3012
3013                                  ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3014                                  ;STEP INTO END OF SECTOR DEAD BAND
3015                                  ;EBL IS NOW ASSERTED
3016
3017  013546  012737  000010  001202  MRD4:   MOV     #8.,REPT
3018  013554  104422                  1$:     MRCLK                   ;CLOCK MR REG
3019  013556  104420                          MRCK                    ;CHECK MR REG
3020  013560  003601                          3601                    ;TO EQUAL
3021  013562  104000                          HLT                     ;3601
3022  013564  104422                          MRCLK                   ;CLOCK MR REG
3023  013566  104420                          MRCK                    ;CHECK MR
3024  013570  153601                          153601                  ;REG TO
3025  013572  104000                          HLT                     ;EQUAL 153601
3026  013574  005337  001202                  DEC     REPT            ;DONE YET?
3027  013600  001365                          BNE     1$              ;NO
3028
3029                                  ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3030                                  ;SHOULD GET STROBE BUFFER
3031
3032  013602  104422                          MRCLK                   ;CLOCK MR REG
3033  013604  104420                          MRCK                    ;CHECK MR
3034  013606  007601                          7601                    ;REG TO
3035  013610  104000                          HLT                     ;EQUAL 7601
```

```
                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.
                                
013612  104422                  MRD5:   MRCLK                   ;CLOCK MR REG
013614  022777  004270  165256          CMP     #4270,@RSCS1    ;ANY ERRORS?
013622  001401                        BEQ     1$              ;NO
013624  104054                        HLT     !DA!DS!WC
013626  005777  165252          1$:     TST     @RSWC           ;DID WC GO TO 0
013632  001401                        BEQ     .+4             ;YES
013634  104010                        HLT     !WC             ;WC REG SHOULD=0
013636  022777  000001  165244        CMP     #1,@RSDA        ;DOES RSDA=1
013644  001401                        BEQ     .+4             ;YES
013646  104004                        HLT     !DA             ;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
                                
013650  042737  000040  001166  MRD6:   BIC     #BIT5,ONCEE     ;CLEAR MAINT CLK CNT
013656  012701  027566                MOV     #INBUF,GOOD     ;GET STARTING LOC OF EXPECTED DATA
013662  012700  030366                MOV     #OUTBUF,BAD     ;GET STARTING LOC OF DATA "READ" FROM DISK
013666  012737  000012  001202        MOV     #12,REPT        ;SET UP ERROR COUNTER
013674  012705  000201                MOV     #201,R5         ;COMPARE 1 SECTOR
013700  005305                  3$:     DEC     R5              ;DONE WITH SECTOR
013702  001445                        BEQ     2$              ;YES GET OUT
013704  022021                        CMP     (BAD)+,(GOOD)+  ;IS DATA CORRECT?
013706  001774                        BEQ     3$              ;YES
013710  032777  010000  165110        BIT     #BIT12,@SWR     ;TYPE ALL ERRORS?
013716  001003                        BNE     1$              ;YES
013720  005337  001202                DEC     REPT            ;TYPED OUT 10 ERRORS YET?
013724  001434                        BEQ     2$              ;YES GET OUT
013726  024041                  1$:     CMP     -(BAD),-(GOOD)  ;GET ERROR
013730  010003                        MOV     BAD,R3          ;SAVE CONTENTS
013732  010104                        MOV     GOOD,R4         ;OF GOOD AND BAD
013734  011300                        MOV     (R3),BAD        ;GET DATA AND PLACE
013736  011401                        MOV     (R4),GOOD       ;IN GOOD AND BAD
013740  104000                        HLT                     ;TYPE OUT ERROR
013742  010300                        MOV     R3,BAD          ;PUT ADDRESS BACK
013744  010401                        MOV     R4,GOOD         ;INTO GOOD AND BAD
013746  010037  001214                MOV     BAD,WORK
013752  032777  020000  165046        BIT     #BIT13,@SWR     ;TYPE OUT MESSAGE?
013760  001014                        BNE     4$              ;NO
013762  104402  013766                TYPE    ,.+2            ;.ASCIZ "BAD ADDRESS= "
014004  013746  001214                MOV     WORK,-(6)       ;PUT WORK ON STACK
014010  104406                        TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
014012  022021                  4$:     CMP     (BAD)+,(GOOD)+
014014  000731                        BR      3$
014016                          2$:                             ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 56                    MAINTENANCE MODE DATA WRITE CHECK TEST
                                ;*****************************************************************************
014016  104400                  TST56:  SCOPE

                                ;MODULE TESTED: M7771, M7753, M7751
                                ;A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION.
                                ;WITHIN THE RS04, A WRITE CHECK FUNCTION IS IDENTICAL TO A
                                ;READ FUNCTION.

014020  104414                  MRWCK:  CLRDK                   ;CLEAR DRIVE REGISTERS
014022  052737  000040  001166          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
014030  042737  147716  001166          BIC     #147716,ONCEE   ;CLEAR ALL OTHER FLAG BITS
014036  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
014040  104420                          MRCK                    ;CHECK MR REG
014042  022701                          22701                   ;TO EQUAL 22701
014044  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES

014046  012737  000004  001144          MOV     #4,FLAG2        ;SET WC FLAG FOR CLKR1 ROUTINE

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 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 (110 WORDS)

014054  012702  027566                  MOV     #INBUF,R2       ;GET LOCATION OF INBUF
014060  005022                          CLR     (R2)+           ;CLEAR 1ST LOCATION
014062  012722  177777                  MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
014066  005003                          CLR     R3              ;CLEAR WORK LOC TO GENERATE
014070  000261                          SEC                     ;A PATTERN OF FLOATING ONES
014072  006103                  1$:     ROL     R3              ;GET PATTERN
014074  103402                          BCS     2$              ;DONE GET OUT
014076  010322                          MOV     R3,(R2)+        ;FILL BUFFER
014100  000774                          BR      1$              ;CONT
014102  012703  000156          2$:     MOV     #110.,R3        ;FILL REMAINING PORTION OF
014106  012704  146314                  MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
014112  010422                  3$:     MOV     R4,(R2)+        ;LOAD BUFFER
014114  005303                          DEC     R3              ;DONE YET
014116  001375                          BNE     3$              ;NO

                                ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0

014120  012777  027566  164760          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
014126  012777  177600  164750          MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
014134  012777  000051  164736          MOV     #51,@RSCS1      ;LOAD WRITE CHECK COMMAND
014142  104446                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
014144  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
014146  104450                          SPASS                   ;CLOCK MR SECTOR PULSE = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
014150  104430                                MRIND
014152  104420                                MRCK                    ;CHECK MR REG TO EQUAL
014154  022701                                22701                   ;22701 FOR A
014156  104000                                HLT                     ;WRITE CHECK COMMAND

                                        ;STEP THRU RESYNC PERIOD

014160  012737  001000  001202                MOV     #512.,REPT
014166  052737  000040  001166                BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERRORS OCCUR
014174  104422                        MRWCK1: MRCLK                   ;CLOCK MR REG
014176  104420                                MRCK                    ;CHECK FOR
014200  072701                                72701                   ;CORRECT DATA
014202  104000                                HLT                     ;MR=BAD GOOD=CORRECT DATA
014204  104422                                MRCLK                   ;CLOCK MR REG
014206  104420                                MRCK                    ;CHECK FOR
014210  022701                                22701                   ;CORRECT DATA
014212  104000                                HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
014214  005337  001202                        DEC     REPT            ;FINISH LOOPING
014220  001365                                BNE     MRWCK1          ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
014222  104422                                MRCLK                   ;CLOCK MR REG
014224  104420                                MRCK                    ;MR SHOULD
014226  072301                                72301                   ;EQUAL 72301
014230  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
014232  104422                                MRCLK                   ;CLOCK MR REG
014234  104420                                MRCK                    ;CHECK MR
014236  022301                                22301                   ;TO EQUAL 22301
014240  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--

014242  012737  000107  001202                MOV     #71.,REPT
014250  104422                        MRWCK2: MRCLK                   ;CLOCK MR REG
014252  104420                                MRCK                    ;CHECK MR REG
014254  073701                                73701                   ;TO EQUAL 73701
014256  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
014260  104422                                MRCLK                   ;CLOCK MR REG
014262  104420                                MRCK                    ;CHECK MR REG
014264  023701                                23701                   ;TO EQUAL 23701
014266  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
014270  005337  001202                        DEC     REPT            ;DONE YET
014274  001365                                BNE     MRWCK2          ;NO LOOP
014276  104422                                MRCLK                   ;CLOCK MR REG
014300  104420                                MRCK                    ;CHECK MR REG
014302  073701                                73701                   ;TO EQUAL 73701
014304  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;READ SYNC"1"
                                        
014306 012777 000055 164610             MOV     #55,@RSMR
014314 012777 000045 164602             MOV     #45,@RSMR
014322 104420                           MRCK                    ;CHECK MR FOR
014324 023745                           23745                   ;FOR CORRECT
014326 104000                           HLT                     ;ANS IS IN GOOD
014330 012777 000055 164566             MOV     #55,@RSMR
014336 012777 000045 164560             MOV     #45,@RSMR
014344 104420                           MRCK
014346 173745                           173745
014350 104000                           HLT

                                        ;READ DATA
014352 005037 001224            MRWCK3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
014356 012705 027566                    MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
014362 162705 000002                    SUB     #2,R5
014366 012737 000025 001204             MOV     #21.,REPT1      ;SETUP COUNTER FOR 1ST SB BIT
014374 012737 002200 001202             MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;128 WORDS-9X128=1152
                                                                ;2 CLOCKS PER 2 BITS OF DATA
014402 104444                   1$:     RBIT                    ;GET 2 DATA BITS
014404 104440                           CLKR1                   ;CLOCK MR
014406 104000                           HLT                     ;MR REG NOT CORRECT
014410 104442                           CLKR2                   ;CLOCK MR REG
014412 104000                           HLT                     ;MR REG NOT CORRECT
014414 005337 001202                    DEC     REPT            ;DONE WITH DATA BUFFER YET?
014420 001370                           BNE     1$              ;NO
014422 032737 000400 001166     2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
014430 001030                           BNE     3$              ;YES
014432 052737 000400 001166             BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
014440 013737 001204 001172             MOV     REPT1,SAVEE     ;SAVE REPT1
014446 004737 024112                    JSR     PC,GENCRC       ;GENERATE CRC WORD
                                                                ;AND LEAVE IN LOC "WORK"
014452 012702 027566                    MOV     #INBUF,R2
014456 013737 001172 001204             MOV     SAVEE,REPT1     ;RESTORE REPT1
014464 062702 000400                    ADD     #400,R2         ;STORE CRC WORD TO BE READ
014470 013712 001214                    MOV     WORK,@R2        ;AT END OF INBUF TABLE
014474 010205                           MOV     R2,R5
014476 162705 000002                    SUB     #2,R5
014502 012737 000011 001202             MOV     #9.,REPT        ;SETUP TO TRANSFER 1 WD
014510 000734                           BR      1$              ;TRANSFER CRC WD
```

```
3227  014512  104422                    3$:     MRCLK                   ;CLOCK MR REG
3228  014514  104420                            MRCK                    ;CHECK MR REG
3229  014516  003701                            3701                    ;TO EQUAL
3230  014520  104000                            HLT                     ;3701
3231  014522  104422                            MRCLK                   ;CLOCK MR REG
3232  014524  104420                            MRCK                    ;CHECK MR
3233  014526  153701                            153701                  ;TO EQUAL
3234  014530  104000                            HLT                     ;153701
3235  014532  104422                            MRCLK                   ;CLOCK MR REG
3236  014534  104420                            MRCK                    ;CHECK MR
3237  014536  007701                            7701                    ;TO EQUAL
3238  014540  104000                            HLT                     ;7701
3239  014542  104422                            MRCLK                   ;CLOCK MR REG
3240  014544  104420                            MRCK                    ;CHECK MR
3241  014546  153701                            153701                  ;TO EQUAL
3242  014550  104000                            HLT                     ;153701
3243
3244                                    ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3245                                    ;STEP INTO END OF SECTOR DEAD BAND
3246                                    ;EBL IS NOW ASSERTED
3247
3248  014552  012737  000010  001202  MRWCK4: MOV     #8.,REPT
3249  014560  104422                    1$:     MRCLK                   ;CLOCK MR REG
3250  014562  104420                            MRCK                    ;CHECK MR REG
3251  014564  003701                            3701                    ;TO EQUAL
3252  014566  104000                            HLT                     ;3601
3253  014570  104422                            MRCLK                   ;CLOCK MR REG
3254  014572  104420                            MRCK                    ;CHECK MR
3255  014574  153701                            153701                  ;REG TO
3256  014576  104000                            HLT                     ;EQUAL 153601
3257  014600  005337  001202                    DEC     REPT            ;DONE YET?
3258  014604  001365                            BNE     1$              ;NO
3259
3260                                    ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3261                                    ;SHOULD GET STROBE BUFFER
3262
3263  014606  104422                            MRCLK                   ;CLOCK MR REG
3264  014610  104420                            MRCK                    ;CHECK MR
3265  014612  007701                            7701                    ;REG TO
3266  014614  104000                            HLT                     ;EQUAL 7601
3267
3268                                    ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3269                                    ;SHOULD COMPLETE TRANSFER.
3270
3271  014616  104422                    MRWCK5: MRCLK
3272  014620  022777  004250  164252            CMP     #4250,@RSCS1    ;ANY ERRORS?
3273  014626  001401                            BEQ     1$              ;NO
3274  014630  104054                            HLT     !DA!DS!WC
3275  014632  005777  164246            1$:     TST     @RSWC           ;DID WC GO TO 0
3276  014636  001401                            BEQ     .+4             ;YES
3277  014640  104010                            HLT     !WC             ;WC REG SHOULD=0
3278  014642  022777  000001  164240            CMP     #1,@RSDA        ;DOES RSDA=1
3279  014650  001401                            BEQ     .+4             ;YES
3280  014652  104004                            HLT     !DA             ;RSDA SHOULD=1
```

```
                        ;*************************************************************************
                        ;TEST 57                MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)
                        ;*************************************************************************
014654  104400          TST57:  SCOPE

                        ;MODULES TESTED: M7753>
                        ;THE RS04 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.

014656  012737  000040  001144          MOV     #40,FLAG2       ;CLEAR TST FLAG
014664  104414  MRCRC:  CLRDK           ;CLEAR DRIVE REGISTERS
014666  052737  000040  001166          BIS     #BIT5,ONCEE     ;TYPE CLOCK COUNT IF ERROR OCCURS
014674  042737  147716  001166          BIC     #147716,ONCEE   ;CLEAR ALL OTHER FLAG BITS
014702  104430          MRIND           ;SEND INDEX PULSE TO MR REG
014704  104420          MRCK            ;CHECK MR REG
014706  022701          22701           ;TO EQUAL 22701
014710  104424          MRINT           ;INIT MAINT MODE (CLEAR MRSP)
                                        ;BY SENDING 2 CLOCK PULSES
014712  032737  000020  001144          BIT     #BIT4,FLAG2     ;FIRST TIME THROUGH
014720  001023          BNE     3$      ;NO
014722  012737  000001  001172          MOV     #1,SAVEE        ;LOAD 1ST CRC WORD

                        ;FILL MEMORY DATA BUFFER (INBUF) WITH 1 SECTOR. CREATE BUFFER
                        ;WITH 144 WORDS OF 16 BITS WHICH EQUALS THE NO. OF BITS IN 128 18 BITS WORDS
                        ;DATA BUFFER CONTAINS 14 WORDS OF ZEROS
                        ;                     A WORD OF 12
                        ;                     A WORD OF 20000
                        ;                     128 WORDS OF ZEROS
                        ;
                        ;NOTE:
                        ;IN THIS TEST, ALL 18 BITS OF THE RS04 DATA
                        ;WORD MUST BE MANIPULATED. HENCE, A TABLE
                        ;CONTAINING 2304 BITS (128 X 18) IS REQUIRED
                        ;INSTEAD OF A TABLE CONTAINING 128 WORDS.

014730  012702  027566          MOV     #INBUF,R2       ;GET LOCATION OF INBUF
014734  012703  000016          MOV     #14.,R3         ;SETUP COUNTER
014740  005022  1$:     CLR     (R2)+           ;TO CLEAR THE
014742  005303          DEC     R3              ;FIRST 14
014744  001375          BNE     1$              ;WORDS
014746  012722  000012          MOV     #12,(R2)+       ;LOAD A 12
014752  012722  020000          MOV     #20000,(R2)+    ;LOAD A 20000
014756  012703  000200          MOV     #128.,R3        ;SETUP COUNTER
014762  005022  2$:     CLR     (R2)+           ;TO CLEAR THE
014764  005303          DEC     R3              ;REMAINING WORDS
014766  001375          BNE     2$              ;FOR THAT SECTOR
```

```
                                        ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0

015770  012777  030366  164110  3$:     MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
015776  012777  177600  164100          MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
016004  012777  000071  164066          MOV     #71,@RSCS1      ;LOAD READ COMMAND
016012  012702  000200                  MOV     #200,R2
016016  012703  030366                  MOV     #OUTBUF,R3
016022  052737  000020  001144          BIS     #BIT4,FLAG2     ;SET 1ST TIME THROUGH FLAG
016030  005023                  4$:     CLR     (R3)+
016032  005302                          DEC     R2
016034  001375                          BNE     4$
016036  104446                          GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
016040  104220                          HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
016042  104450                          SPASS                   ;CLOCK MR REG SP = 1

                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
016044  104430                          MRIND
016046  104420                          MRCK                    ;CHECK MR REG TO EQUAL
016050  022601                          22601                   ;22601 FOR A
016052  104000                          HLT                     ;READ COMD

                                        ;STEP THRU RESYNC PERIOD

016054  012737  001000  001202          MOV     #512.,REPT
016062  052737  000040  001166          BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
016070  104422          MRCRC1: MRCLK   ;CLOCK MR REG
016072  104420                  MRCK    ;CHECK FOR
016074  072601                  72601   ;CORRECT DATA
016076  104000                  HLT     ;MR=BAD GOOD=CORRECT DATA
016100  104422                  MRCLK   ;CLOCK MR REG
016102  104420                  MRCK    ;CHECK FOR
016104  022601                  22601   ;CORRECT DATA
016106  104000                  HLT     ;ERROR WHILE CLOCKING THROUGH RESYNC
016110  005337  001202          DEC     REPT            ;FINISH LOOPING
016114  001365                  BNE     MRCRC1          ;THROUGH RESYNC PERIOD

                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                ;SP=0 EQUALS SECTOR PULSE
016116  104422                  MRCLK                   ;CLOCK MR REG
016120  104420                  MRCK                    ;MR SHOULD
016122  072201                  72201                   ;EQUAL 72201
016124  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
016126  104422                  MRCLK                   ;CLOCK MR REG
016130  104420                  MRCK                    ;CHECK MR
016132  022201                  22201                   ;TO EQUAL 22201
016134  104000                  HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--

015136  012737  000107  001202          MOV     #71.,REPT
015144  104422                MRCRC2:   MRCLK                   ;CLOCK MR REG
015146  104420                          MRCK                    ;CHECK MR REG
015150  073601                          73601                   ;TO EQUAL 73601
015152  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
015154  104422                          MRCLK                   ;CLOCK MR REG
015156  104420                          MRCK                    ;CHECK MR REG
015160  023601                          23601                   ;TO EQUAL 23601
015162  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
015164  005337  001202                  DEC     REPT            ;DONE YET
015170  001365                          BNE     MRCRC2          ;NO LOOP
015172  104422                          MRCLK                   ;CLOCK MR REG
015174  104420                          MRCK                    ;CHECK MR REG
015176  073601                          73601                   ;TO EQUAL 73601
015200  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;READ SYNC"1"

015202  012777  000055  163714          MOV     #55,@RSMR
015210  012777  000045  163706          MOV     #45,@RSMR
015216  104420                          MRCK                    ;CHECK MR REG
015220  023645                          23645                   ;FOR CORRECT
015222  104000                          HLT                     ;ANS IS IN GOOD
015224  012777  000055  163672          MOV     #55,@RSMR
015232  012777  000045  163664          MOV     #45,@RSMR
015240  104420                          MRCK
015242  173645                          173645
015244  104000                          HLT

                                        ;READ DATA
015246  005C37  001224        MRCRC3:   CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
015252  012705  027566                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
015256  162705  000002                  SUB     #2,R5
015262  012737  000025  001204          MOV     #21.,REPT1      ;SETUP COUNTER FOR 1ST SB BIT
015270  012737  002200  001202          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;128 WORDS-9X128=1152
                                                                ;2 CLOCKS PER 2 BITS OF DATA
015276  104444                1$:       RBIT                    ;GET 2 DATA BITS
015300  104440                          CLKR1                   ;CLOCK MR
015302  104000                          HLT                     ;MR REG NOT CORRECT
015304  104442                          CLKR2                   ;CLOCK MR REG
015306  104000                          HLT                     ;MR REG NOT CORRECT
015310  005337  001202                  DEC     REPT            ;DONE WITH DATA BUFFER YET?
015314  001370                          BNE     1$              ;NO
```

```
 3426  015316  032737  000400  001166  2$:   BIT   #BIT8,ONCEE    ;DID WE ALREADY DO CRC?
 3427  015324  001020                        BNE   3$             ;YES
 3428  015326  052737  000400  001166        BIS   #BIT8,ONCEE    ;NO SET CRC FLAG
 3429  015334  012702  027566                MOV   #INBUF,R2      ;MOVE CRC
 3430  015340  062702  000440                ADD   #440,R2        ;WORD TO END OF
 3431  015344  013712  001172        4$:     MOV   SAVEE,@R2      ;INBUF TABLE
 3432  015350  010205                5$:     MOV   R2,R5          ;GET CRC WORD
 3433  015352  162705  000002                SUB   #2,R5
 3434  015356  012737  000011  001202        MOV   #9.,REPT       ;SETUP TO TRANSFER 1 WD
 3435  015364  000744                        BR    1$             ;TRANSFER CRC WD
 3436  015366  104422                3$:     MRCLK                ;CLOCK MR REG
 3437  015370  104420                        MRCK                 ;CHECK MR REG
 3438  015372  003601                        3601                 ;TO EQUAL
 3439  015374  104000                        HLT                  ;3601
 3440  015376  104422                        MRCLK                ;CLOCK MR REG
 3441  015400  104420                        MRCK                 ;CHECK MR
 3442  015402  153601                        153601               ;TO EQUAL
 3443  015404  104000                        HLT                  ;153601
 3444  015406  104422                        MRCLK                ;CLOCK MR REG
 3445  015410  104420                        MRCK                 ;CHECK MR
 3446  015412  007601                        7601                 ;TO EQUAL
 3447  015414  104000                        HLT                  ;7601
 3448  015416  104422                        MRCLK                ;CLOCK MR REG
 3449  015420  104420                        MRCK                 ;CHECK MR
 3450  015422  153601                        153601               ;TO EQUAL
 3451  015424  104000                        HLT                  ;153601
```

```
                                        ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
                                        ;STEP INTO END OF SECTOR DEAD BAND
                                        ;EBL IS NOW ASSERTED

015426  012737  000010  001202  MRCRC4: MOV     #8.,REPT

015434  104422                  1$:     MRCLK                   ;CLOCK MR REG
015436  104420                          MRCK                    ;CHECK MR REG
015440  003601                          3601                    ;TO EQUAL
015442  104000                          HLT                     ;3601
015444  104422                          MRCLK                   ;CLOCK MR REG
015446  104420                          MRCK                    ;CHECK MR
015450  153601                          153601                  ;REG TO
015452  104000                          HLT                     ;EQUAL 153601
015454  005337  001202                  DEC     REPT            ;DONE YET?
015460  001365                          BNE     1$              ;NO

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD GET STROBE BUFFER

015462  104422                          MRCLK                   ;CLOCK MR REG
015464  104420                          MRCK                    ;CHECK MR
015466  007601                          7601                    ;REG TO
015470  104000                          HLT                     ;EQUAL 7601

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.

015472  104422          MRCRC5: MRCLK                           ;CLOCK MR REG
015474  022777  004270  163376          CMP     #4270,@RSCS1    ;ANY ERRORS?
015502  001401                          BEQ     1$              ;NO
015504  104054                          HLT     !DA!DS!WC
015506  005777  163372          1$:     TST     @RSWC           ;DID WC GO TO 0
015512  001401                          BEQ     .+4             ;YES
015514  104010                          HLT     !WC             ;WC REG SHOULD=0
015516  006137  001172                  ROL     SAVEE           ;GET NEXT CRC WORD
015522  103404                          BCS     2$              ;DONE - BRANCH
015524  004737  026270                  JSR     PC,MDATA        ;SHIFT DATA PATTERN
015530  000137  014664                  JMP     MRCRC           ;RESTART TEST WITH NEW DATA PATTERN
015534                          2$:                             ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 60                    MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)
                                ;*****************************************************************************
015534  104400                  TST60:  SCOPE
                                ;MODULE TESTED M7753
                                ;THIS TEST IS SIMILAR TO CRC TEST 1 EXECPT 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.
015536  012737  000040  001144          MOV     #40,FLAG2       ;CLEAR TST FLAG
015544  104414          MRDCK:  CLRDK                           ;CLEAR DRIVE REGISTERS
015546  052737  000040  001166          BIS     #BIT5,ONCEE     ;TYPE CLOCK COUNT IF ERROR OCCURS
015554  042737  147716  001166          BIC     #147716,ONCEE   ;CLEAR ALL OTHER FLAG BITS
015562  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
015564  104420                          MRCK                    ;CHECK MR REG
015566  022701                          22701                   ;TO EQUAL 22701
015570  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                                ;BY SENDING 2 CLOCK PULSES
015572  032737  000020  001144          BIT     #BIT4,FLAG2     ;FIRST TIME THROUGH
015600  001023                          BNE     3$              ;NO
015602  012737  000001  001172          MOV     #1,SAVEE        ;LOAD 1ST CRC WORD
                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR) CREATE BUFFER
                                ;WITH 144 WORDS OF 16 BITS WHICH = THE NO. OF BITS IN 128 18 BIT WORDS
                                ;DATA BUFFER CONTAINS 15 WORDS OF ZEROS
                                ;                     A WORD OF 1
                                ;                     A WORD OF 42000
                                ;                     127 WORDS OF ZEROS
015610  012702  027566                  MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
015614  012703  000017                  MOV     #15.,R3         ;SETUP COUNTER
015620  005022          1$:     CLR     (R2)+                   ;TO CLEAR THE
015622  005303                  DEC     R3                      ;FIRST 15
015624  001375                  BNE     1$                      ;WORDS
015626  012722  000001                  MOV     #1,(R2)+        ;LOAD A 1
015632  012722  042000                  MOV     #42000,(R2)+    ;LOAD A 42000
015636  012703  000177                  MOV     #127.,R3        ;SETUP COUNTER
015642  005022          2$:     CLR     (R2)+                   ;TO CLEAR THE
015644  005303                  DEC     R3                      ;REMAINING WORDS
015646  001375                  BNE     2$                      ;FOR THAT SECTOR
                                ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
015650  012777  030366  163230  3$:     MOV     #OUTBUF,@RSBA   ;LOAD BUS ADDR REG
015656  012777  177600  163220          MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
015664  012777  000071  163206          MOV     #71,@RSCS1      ;LOAD READ COMMAND
015672  012702  000200                  MOV     #200,R2
015676  012703  030366                  MOV     #OUTBUF,R3
015702  052737  000020  001144          BIS     #BIT4,FLAG2     ;SET 1ST TIME THROUGH FLAG
015710  005023          4$:     CLR     (R3)+
015712  005302                  DEC     R2
015714  001375                  BNE     4$
015716  104446                  GETSP                           ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
015720  104220                  HLT     !MR                     ;COULD NOT SET SECTOR PULSE (0)
015722  104450                  SPASS                           ;CLOCK MR REG SP = 1
```

```
3546                                    ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3547  015724  104430                            MRIND
3548  015726  104420                            MRCK                    ;CHECK MR REG TO EQUAL
3549  015730  022601                            22601                   ;22601 FOR A
3550  015732  104000                            HLT                     ;READ COMD
3551
3552                                    ;STEP THRU RESYNC PERIOD
3553
3554  015734  012737  001000  001202            MOV     #512.,REPT
3555  015742  052737  000040  001166            BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
3556  015750  104422                    MRDCK1: MRCLK                   ;CLOCK MR REG
3557  015752  104420                            MRCK                    ;CHECK FOR
3558  015754  072601                            72601                   ;CORRECT DATA
3559  015756  104000                            HLT                     ;MR=BAD GOOD=CORRECT DATA
3560  015760  104422                            MRCLK                   ;CLOCK MR REG
3561  015762  104420                            MRCK                    ;CHECK FOR
3562  015764  022601                            22601                   ;CORRECT DATA
3563  015766  104000                            HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
3564  015770  005337  001202                    DEC     REPT            ;FINISH LOOPING
3565  015774  001365                            BNE     MRDCK1          ;THROUGH RESYNC PERIOD
3566
3567                                    ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3568                                    ;SP=0 EQUALS SECTOR PULSE
3569  015776  104422                            MRCLK                   ;CLOCK MR REG
3570  016000  104420                            MRCK                    ;MR SHOULD
3571  016002  072201                            72201                   ;EQUAL 72201
3572  016004  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
3573  016006  104422                            MRCLK                   ;CLOCK MR REG
3574  016010  104420                            MRCK                    ;CHECK MR
3575  016012  022201                            22201                   ;TO EQUAL 22201
3576  016014  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                        ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--

016016  012737  000107  001202  MRDCK2: MOV     #71.,REPT
016024  104422                  MRDCK2: MRCLK                   ;CLOCK MR REG
016026  104420                          MRCK                    ;CHECK MR REG
016030  073601                          73601                   ;TO EQUAL 73601
016032  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
016034  104422                          MRCLK                   ;CLOCK MR REG
016036  104420                          MRCK                    ;CHECK MR REG
016040  023601                          23601                   ;TO EQUAL 23601
016042  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
016044  005337  001202                  DEC     REPT            ;DONE YET
016050  001365                          BNE     MRDCK2          ;NO LOOP
016052  104422                          MRCLK                   ;CLOCK MR REG
016054  104420                          MRCK                    ;CHECK MR REG
016056  073601                          73601                   ;TO EQUAL 73601
016060  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;READ SYNC"1"

016062  012777  000055  163034          MOV     #55,@RSMR
016070  012777  000045  163026          MOV     #45,@RSMR
016076  104420                          MRCK                    ;CHECK MR REG
016100  023645                          23645                   ;TO EQUAL
016102  104000                          HLT                     ;CORRECT ANS IN GOOD
016104  012777  000055  163012          MOV     #55,@RSMR
016112  012777  000045  163004          MOV     #45,@RSMR
016120  104420                          MRCK
016122  173645                          173645
016124  104000                          HLT

                                        ;READ DATA
016126  005037  001224          MRDCK3: CLR     WORK3           ;CLEAR CLOCK COUNT FOR DATA WD
016132  012705  027566                  MOV     #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
016136  162705  000002                  SUB     #2,R5
016142  012737  000025  001204          MOV     #21.,REPT1      ;SETUP COUNTER FOR 1ST SB BIT
016150  012737  002200  001202          MOV     #1152.,REPT     ;SETUP COUNTER TO TRANSFER
                                                                ;128 WORDS-9X128=1152
                                                                ;2 CLOCKS PER 2 BITS OF DATA
016156  104444                  1$:     RBIT                    ;GET 2 DATA BITS
016160  104440                          CLKR1                   ;CLOCK MR
016162  104000                          HLT                     ;MR REG NOT CORRECT
016164  104442                          CLKR2                   ;CLOCK MR REG
016166  104000                          HLT                     ;MR REG NOT CORRECT
016170  005337  001202                  DEC     REPT            ;DONE WITH DATA BUFFER YET?
016174  001370                          BNE     1$              ;NO
```

```
3623  016176  032737  000400  001166  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
3624  016204  001020                          BNE     3$              ;YES
3625  016206  052737  000400  001166          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
3626  016214  012702  027566                  MOV     #INBUF,R2       ;MOVE CRC
3627  016220  062702  000440                  ADD     #440,R2         ;WORD TO END OF
3628  016224  012712  000000          4$:     MOV     #0,@R2          ;INBUF TABLE
3629  016230  010205                  5$:     MOV     R2,R5           ;GET CRC WORD
3630  016232  162705  000002                  SUB     #2,R5
3631  016236  012737  000011  001202          MOV     #9.,REPT        ;SETUP TO TRANSFER 1 WD
3632  016244  000744                          BR      1$              ;TRANSFER CRC WD
3633  016246  104422                  3$:     MRCLK                   ;CLOCK MR REG
3634  016250  104420                          MRCK                    ;CHECK MR REG
3635  016252  003601                          3601                    ;TO EQUAL
3636  016254  104000                          HLT                     ;3601
3637  016256  104422                          MRCLK                   ;CLOCK MR REG
3638  016260  104420                          MRCK                    ;CHECK MR
3639  016262  153601                          153601                  ;TO EQUAL
3640  016264  104000                          HLT                     ;153601
3641  016266  104422                          MRCLK                   ;CLOCK MR REG
3642  016270  104420                          MRCK                    ;CHECK MR
3643  016272  007601                          7601                    ;TO EQUAL
3644  016274  104000                          HLT                     ;7601
3645  016276  104422                          MRCLK                   ;CLOCK MR REG
3646  016300  104420                          MRCK                    ;CHECK MR
3647  016302  153601                          153601                  ;TO EQUAL
3648  016304  104000                          HLT                     ;153601
```

```
                                        ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
                                        ;STEP INTO END OF SECTOR DEAD BAND
                                        ;EBL IS NOW ASSERTED

016306  012737  000010  001202  MRDCK4: MOV     #8.,REPT

016314  104422                  1$:     MRCLK                   ;CLOCK MR REG
016316  104420                          MRCK                    ;CHECK MR REG
016320  003601                          3601                    ;TO EQUAL
016322  104000                          HLT                     ;3601
016324  104422                          MRCLK                   ;CLOCK MR REG
016326  104420                          MRCK                    ;CHECK MR
016330  153601                          153601                  ;REG TO
016332  104000                          HLT                     ;EQUAL 153601
016334  005337  001202                  DEC     REPT            ;DONE YET?
016340  001365                          BNE     1$              ;NO

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD GET STROBE BUFFER

016342  104422                          MRCLK                   ;CLOCK MR REG
016344  104420                          MRCK                    ;CHECK MR
016346  007601                          7601                    ;REG TO
016350  104000                          HLT                     ;EQUAL 7601

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.

016352  104422          MRDCK5: MRCLK                           ;CLOCK MR REG
016354  022777  144270  162516          CMP     #144270,@RSCS1  ;IS RSCS1 CORRECT?
016362  001401                          BEQ     1$              ;YES
016364  104054                          HLT     !DA!DS!WC
016366  005777  162512          1$:     TST     @RSWC           ;DID WC GO TO 0
016372  001401                          BEQ     .+4             ;YES
016374  104010                          HLT     !WC             ;WC REG SHOULD=0
016376  022777  100000  162510          CMP     #100000,@RSER   ;DID DCK SET?
016404  001417                          BEQ     3$              ;YES
016406  104050                          HLT     !DS!WC
016410  104402  016414                  TYPE    .+2             ;.ASCIZ <15><12>"DCK DID NOT SET "
016440  004737  022654                  JSR     PC,CRCTYP       ;GET IC THAT FAILED AND TYPE IT
016444  000241          3$:     CLC
016446  006137  001172                  ROL     SAVEE           ;GET NEXT CRC WORD
016452  103404                          BCS     2$              ;DONE - BRANCH
016454  004737  026270                  JSR     PC,MDATA        ;SHIFT DATA PATTERN
016460  000137  015544                  JMP     MRDCK           ;RESTART TEST WITH NEW DATA PATTERN
016464          2$:                     ;DONE
```

```
                        ;*****************************************************************************
                        ;TEST 61                    IGNORE FUNCTION TEST
                        ;*****************************************************************************
016464  104400          TST61:  SCOPE

                        ;MODULE TESTED: M7759, M7770
                        ;PUT THE DISK 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 IS TURN SHOULD
                        ;CAUSE "TRE" AND "SC" TO SET IN RSCS1.

016466  104414          MRIFT:  CLRDK                   ;CLEAR ALL REGISTERS
016470  104430                  MRIND                   ;SEND INDEX PULSE TO MR REG
016472  104420                  MRCK                    ;CHECK MR REG
016474  022701                  22701                   ;TO EQUAL 22701
016476  104424                  MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
016500  012777  177777  162406        MOV     #-1,@RSER       ;SET ERRORS
016506  013777  001162  162402        MOV     UNITSV,@RSAS    ;CLEAR ATA BIT IN RSAS
                                                        ;AND ERROR BITS IN RSCS1
016514  012777  030366  162364        MOV     #OUTBUF,@RSBA   ;LOAD RSBA
016522  012777  177777  162354        MOV     #-1,@RSWC       ;LOAD RSWC
016530  012777  000071  162342        MOV     #71,@RSCS1      ;LOAD READ FUNCTION
016536  032777  000001  162334        BIT     #BIT0,@RSCS1    ;IS "GO" BIT ZERO?
016544  001401                  BEQ     1$              ;YES
016546  104140                  HLT     !DS!AS          ;"GO" BIT IN RSCS1 SHOULD NOT
                                                        ;LOAD IF ERRORS ARE PRESENT IN THE DRIVE
016550  012737  177777  001214  1$:     MOV     #177777,WORK    ;SETUP TIMEOUT FOR MXF ERROR
016556  005337  001214          5$:     DEC     WORK
016562  000240                  NOP
016564  000240                  NOP
016566  001373                  BNE     5$
016570  017700  162306                  MOV     @RSCS2,BAD      ;CHECK RSCS2 FOR MXF
016574  012701  001100                  MOV     #1100,GOOD      ;GET CORRECT ANS
016600  053701  001160                  BIS     UNNUM,GOOD      ;FOR RSCS2
016604  020001                  CMP     BAD,GOOD        ;IS RSCS2 CORRECT
016606  001401                  BEQ     2$              ;YES
016610  104000                  HLT                     ;BAD=RSCS2 GOOD=CORRECT ANS
                                                        ;MXF SHOULD BE SET IN RSCS2
                                                        ;FOR A READ WAS ISSUED
                                                        ;WITH ERROR BITS SET IN RSER.
016612  022777  144270  162260  2$:     CMP     #144270,@RSCS1  ;IS RSCS1 CORRECT?
016620  001401                  BEQ     3$              ;YES
016622  104042                  HLT     !DS!ER          ;SC AND TRE SHOULD BE SET FOR
                                                        ;MXF SHOULD BE SET IN RSCS2
```

```
3740  016624  022777  177777  162252  3$:   CMP   #-1,@RSWC      ;DID RSWC CHANGE?
3741  016632  001401                  BEQ   .+4                  ;NO
3742  016634  104010                  HLT   !WC                  ;WC SHOULD NOT HAVE CHANGED
3743                                                             ;FOR READ SHOULD HAVE NEVER
3744                                                             ;BEEN EXECUTED
3745  016636  022777  030366  162242        CMP   #OUTBUF,@RSBA  ;DID RSBA MOVE
3746  016644  001405                  BEQ   4$                   ;NO
3747  016646  017700  162234              MOV   @RSBA,BAD        ;GET RSBA
3748  016652  012701  030366              MOV   #OUTBUF,GOOD     ;GET CORRECT ANS
3749  016656  104000                  HLT                        ;RSBA=BAD GOOD=CORRECT ANS
3750                                                             ;RSBA MOVED WHEN THE READ
3751                                                             ;COND WAS LOADED WITH ERROR
3752                                                             ;BITS SET IN RSER, READ COMD SHOULD
3753                                                             ;NEVER HAVE BEEN EXECUTED AND RSBA
3754                                                             ;SHOULD NOT HAVE MOVED.
3755  016660  104414              4$:   CLRDK                    ;CLEAR ALL REGISTERS
3756  016662  017700  162214              MOV   @RSCS2,BAD       ;GET RSCS2
3757  016666  012701  000100              MOV   #100,GOOD        ;SETUP CORRECT
3758  016672  053701  001160              BIS   UNNUM,GOOD       ;ANS
3759  016676  020100                  CMP   GOOD,BAD             ;IS CS2 CORRECT?
3760  016700  001401                  BEQ   .+4                  ;YES
3761  016702  104000                  HLT                        ;MXF SHOULD HAVE CLEARED IN
3762                                                             ;RSCS2 WITH THE "CLR" BIT BEING
3763                                                             ;SET IN RSCS2.
```

```
                                ;*****************************************************************************
                                ;TEST 62                    INVALID ADDRESS ERROR (IAE) TEST
                                ;*****************************************************************************
016704  104400                  TST62:  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).
016706  042737  000040  001166          BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
016714  012702  004000                  MOV     #4000,R2        ;LOAD R2 WITH INVALID ADDR
016720  012737  016726  001010          MOV     #4$,LAD         ;LOOP TO HERE ON ERROR
016726  104416                  4$:     MRDMD                   ;PUT DRIVE IN MAINT MODE
016730  104420                          MRCK                    ;CHECK MAINT REG
016732  022701                          22701
016734  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
016736  032737  000004  001166          BIT     #BIT2,ONCEE     ;LOOPING ON ERRORS>
016744  001002                          BNE     1$              ;YES
016746  006102                          ROL     R2              ;GET INVALID ADDRESS
016750  103454                          BCS     IADONE          ;DONE FLOATING A ONE YET?
016752  010277  162132          1$:     MOV     R2,@RSDA        ;LOAD RSDA WITH INVALID ADDRESS
016756  012777  000071  162114          MOV     #71,@RSCS1      ;DO A READ TO INVALID ADDR
016764  022777  002000  162122          CMP     #2000,@RSER     ;IS RSER CORRECT?
016772  001404                          BEQ     2$              ;YES
016774  052737  000004  001166          BIS     #BIT2,ONCEE     ;SET ERROR BIT
017002  104044                          HLT     !DS!DA          ;RSER SHOULD=2000 FOR
                                                                ;A READ COMMAND WAS GIVEN
                                                                ;TO AN ILLEGAL ADDRESS
017004  042737  000004  001166  2$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
017012  022777  150600  162072          CMP     #150600,@RSDS   ;DID IAE SET?
017020  001404                          BEQ     3$              ;YES
017022  052737  000004  001166          BIS     #BIT2,ONCEE     ;SET ERROR BIT
017030  104044                          HLT     !DS!DA          ;RSDS SHOULD=150600 FOR
                                                                ;IAE SHOULD BE SET IN RSER
017032  042737  000004  001166  3$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR FLAG
017040  022777  144270  162032          CMP     #144270,@RSCS1  ;DID SC + TRE SET?
017046  001404                          BEQ     5$              ;YES
017050  052737  000004  001166          BIS     #BIT2,ONCEE     ;SET ERROR BIT
017056  104044                          HLT     !DA!DS          ;SC + TRE SHOULD BE SET IN RSCS1
                                                                ;FOR IAE SHOULD BE SET IN RSER
017060  042737  000004  001166  5$:     BIC     #BIT2,ONCEE     ;CLEAR ERROR BIT
017066  104414                          CLRDK                   ;CLEAR ALL ERRORS
017070  005777  162020                  TST     @RSER           ;DID IAE CLEAR?
017074  001401                          BEQ     .+4             ;YES
017076  104040                          HLT     !DS             ;IAE DID NOT CLEAR
017100  000712                          BR      4$              ;CONTINUE
017102                          IADONE:                         ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 63                    OPERATION INCOMPLETE ERROR TEST
                                ;*****************************************************************************
017102  104400                  TST63:  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)

017104  104414          MROPI:  CLRDK                   ;CLEAR ALL DRIVE REGISTERS
017106  013777  030366  161772          MOV     @#OUTBUF,@RSBA  ;SETUP RSBA
017114  012777  177777  161762          MOV     #-1,@RSWC       ;SETUP RSWC

017122  104430                  MRIND                   ;SEND INDEX PULSE TO MR REG
017124  104420                  MRCK                    ;CHECK MAINT REG
017126  022701                  22701                   ;TO EQUAL 22701
017130  104424                  MRINT                   ;INIT MAINT MODE (CLEAR MRSP)

017132  012777  000071  161740          MOV     #71,@RSCS1      ;LOAD A READ COMMAND

017140  104430                  MRIND                   ;ISSUE THREE INDEX
017142  104430                  MRIND                   ;PULSES TO
017144  104430                  MRIND                   ;CAUSE OPI

                                ;NOW CHECK FOR CORRECT ERRORS IN RSER AND RSDS
017146  017700  161742                  MOV     @RSER,BAD       ;GET RSER
017152  012701  020000                  MOV     #20000,GOOD     ;GET CORRECT ANS
017156  020100                          CMP     GOOD,BAD        ;DID OPI SET IN RSER?
017160  001434                          BEQ     1$              ;YES
017162  104402  017166                  TYPE    ,.+2            ;.ASCIZ <15><12>"OPI IN RSER SHOULD SET-3 INDEX PULSES W
017250  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS

017252  022777  150600  161632  1$:     CMP     #150600,@RSDS   ;DID CORRECT ERRORS SET?
017260  001401                          BEQ     2$              ;YES
017262  104040                          HLT     !DS             ;RSDS SHOULD=150600 BECAUSE
                                                                ;OF OPI ERROR IN RSER
017264  022777  144270  161606  2$:     CMP     #144270,@RSCS1  ;DID SC AND TRE SET IN RSCS1?
017272  001401                          BEQ     MROPIA          ;YES
017274  104050                          HLT     !DS!WC          ;SC AND TRE SHOULD SET IN RSCS1
                                                                ;BECAUSE OF ERROR IN RSER
017276  104414          MROPIA: CLRDK                           ;CLEAR ALL ERRORS
017300  005777  161610                  TST     @RSER           ;DID OPI CLEAR IN RSER
017304  001437                          BEQ     1$              ;YES
017306  104402  017312                  TYPE    .+2             ;.ASCIZ <15><12>"OPI IN RSER DID NOT CLEAR BY SETTING CL
017402  104040                          HLT     !DS             ;RSER SHOULD=0
017404  022777  010600  161500  1$:     CMP     #10600,@RSDS    ;DID ERROR BITS CLEAR IN RSDS
                                                                ;BY SETTING CLR BIT IN RSCS2
017412  001401                          BEQ     .+4             ;YES
017414  104040                          HLT     !DS             ;RSDS SHOULD=10600
```

```
                                ;*************************************************************************
                                ;TEST 64                    PARITY ERROR TEST
                                ;*************************************************************************
017416  104400                  TST64:  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.

017420  104414                  MRPAR:  CLRDK                   ;CLEAR ALL REGISTERS
017422  042737  000040  001166          BIC     #BIT5,ONCEE     ;CLEAR CLK CNT FLAG
017430  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
017432  104420                          MRCK                    ;CHECK MAINT TO
017434  022701                          22701                   ;EQUAL 22701
017436  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
017440  052777  000020  161434          BIS     #BIT4,@RSCS2    ;SET THE "PAT" BIT.
017446  012777  000077  161434          MOV     #77,@RSDA       ;BY WRITING INTO THIS REGISTER,
                                                                ;PAR SHOULD SET IN RSER
017454  022777  000010  161432          CMP     #10,@RSER       ;DID PAR SET?
017462  001401                          BEQ     .+4             ;YES
017464  104040                          HLT     !DS             ;"PAR" IN RSER SHOULD BE SET FOR
                                                                ;THE "PAT" BIT WAS SET IN RSCS2
                                                                ;WHEN PROGRAM TRIED TO WRITE INTO RSDA
017466  022777  104200  161404          CMP     #104200,@RSCS1  ;DID PAR CAUSE SC TO SET?
017474  001401                          BEQ     .+4             ;YES
017476  104044                          HLT     !DS!DA          ;SC SHOULD BE SET IN RSCS1 FOR
                                                                ;PAR SHOULD BE SET IN RSER
017500  022777  000077  161402          CMP     #77,@RSDA       ;DID RSDA GET LOADED?
017506  001401                          BEQ     .+4             ;YES
017510  104004                          HLT     !DA             ;RSDA SHOULD=77 FOR PAT
                                                                ;BIT WAS SET WHEN PROGRAM
                                                                ;TRIED TO WRITE INTO RSDA
017512  104414                  CLRDK                           ;CLEAR ALL ERRORS
017514  022777  004200  161356          CMP     #4200,@RSCS1    ;DID ERRORS CLEAR?
017522  001401                          BEQ     .+4             ;YES
017524  104044                          HLT     !DS!DA          ;SC DID NOT CLEAR BY USING
                                                                ;THE "CLR" BIT IN RSCS2
017526  005777  161362                  TST     @RSER           ;DID PAR CLEAR?
017532  001401                          BEQ     .+4             ;YES
017534  104044                          HLT     !DS!DA          ;PAR DID NOT CLEAR BY USING
                                                                ;THE CLR BIT IN RSCS2
```

```
                                ;*******************************************************************************
                                ;TEST 65                    MAINTENANCE MODE INTERRUPT TEST
                                ;*******************************************************************************
017536  104400                  TST65:  SCOPE

                                ;MODULE TESTED M7771
                                ;IN THIS TEST THE INTERRUPT ENABLE BIT IS SET (I.E.).
                                ;A TWO SECTOR WRITE COMMAND IS GIVEN.  AN "RMR"
                                ;ERROR IS CREATED WHILE THE FIRST SECTOR IS BEING WRITTEN
                                ;THIS SHOULD CAUSE THE DRIVE TO INTERRUPT AFTER THE FIRST
                                ;SECTOR IS WRITTEN AND THE TRANSFER TO TERMINATE.

017540  012737  000002  001144          MOV     #2,FLAG2
017546  104414          MREX:   CLRDK                   ;CLEAR DRIVE REGISTERS
017550  012737  000200  177776          MOV     #200,@#PS       ;SETUP FOR INTERRUPT
017556  012706  000500                  MOV     #500,SP
017562  052737  000040  001166          BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
017570  042737  000600  001166          BIC     #600,ONCEE      ;CLEAR FLAG BITS
017576  104430                  MRIND                   ;SEND INDEX PULSE TO MR REG
017600  104420                  MRCK                    ;CHECK MR REG
017602  022701                  22701                   ;TO EQUAL 22701
017604  104424                  MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
                                                        ;BY SENDING 2 CLOCK PULSES

                                ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
                                ;DATA BUFFER WORDS ARE  :A WORD OF ALL 0'S - ALL 1'S
                                ;                       FLOATING 1'S PATTERN (16 WORDS)
                                ;                       A PATTERN OF 146314 (110 WORDS)
017606  012702  027566                  MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
017612  005022                  CLR     (R2)+           ;CLEAR 1ST LOCATION
017614  012722  177777          MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
017620  005003                  CLR     R3              ;CLEAR WORK LOC TO GENERATE
017622  000261                  SEC                     ;A PATTERN OF FLOATING ONES
017624  006103          1$:     ROL     R3              ;GET PATTERN
017626  103402                  BCS     2$              ;DONE GET OUT
017630  010322                  MOV     R3,(R2)+        ;FILL BUFFER
017632  000774                  BR      1$              ;CONT
017634  012703  000156  2$:     MOV     #110.,R3        ;FILL REMAINING PORTION OF
017640  012704  146314          MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
017644  010422          3$:     MOV     R4,(R2)+        ;LOAD BUFFER
017646  005303                  DEC     R3              ;DONE YET?
017650  001375                  BNE     3$              ;NO

                                ;SETUP CONTROLLER TO TRANSFER 256 WORDS OF DATA (2 SECTORS)
017652  012777  020460  161250          MOV     #INTMR,@RSVEC   ;SETUP INTERRUPT VECTOR
017660  012777  000340  161244          MOV     #340,@RSVCPS
017666  012777  027566  161212          MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
017674  012777  177400  161202          MOV     #177400,@RSWC   ;LOAD WORD COUNT REG
017702  012777  000161  161170          MOV     #161,@RSCS1     ;LOAD WRITE COMMAND I/E
017710  104446                  GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
017712  104220                  HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
017714  104450                  SPASS                   ;CLOCK MR REG SP = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
017716  104430                                  MRIND
017720  104420                                  MRCK                    ;CHECK MR REG TO EQUAL
017722  020501                                  20501                   ;20501 FOR A
017724  104000                                  HLT                     ;WRITE COMD

                                        ;STEP THRU RESYNC PERIOD

017726  012737  001000  001202                  MOV     #512.,REPT
017734  052737  000040  001166                  BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
017742  104422                          MREX1:  MRCLK                   ;CLOCK MR REG
017744  104420                                  MRCK                    ;CHECK FOR
017746  070501                                  70501                   ;CORRECT DATA
017750  104000                                  HLT                     ;MR = BAD GOOD = CORRECT DATA
017752  104422                                  MRCLK                   ;CLOCK MR REG
017754  104420                                  MRCK                    ;CHECK FOR
017756  020501                                  20501                   ;CORRECT DATA
017760  104000                                  HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
017762  005337  001202                          DEC     REPT            ;FINISH LOOPING
017766  001365                                  BNE     MREX1           ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
017770  104422                                  MRCLK                   ;CLOCK MR REG
017772  104420                                  MRCK                    ;MR SHOULD
017774  070101                                  70101                   ;EQUAL 70101
017776  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
020000  104422                                  MRCLK                   ;CLOCK MR REG
020002  104420                                  MRCK                    ;CHECK MR
020004  020101                                  20101                   ;TO EQUAL 20101
020006  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 63 DOUBLE MAINT CLOCK OPERATIONS--WRITING PREAMBLE

020010  012737  000077  001202                  MOV     #63.,REPT
020016  104422                          MREX2:  MRCLK                   ;CLOCK MR REG
020020  104420                                  MRCK                    ;CHECK MR REG
020022  071501                                  71501                   ;TO EQUAL 71501
020024  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
020026  104422                                  MRCLK                   ;CLOCK MR REG
020030  104420                                  MRCK                    ;CHECK MR REG
020032  021501                                  21501                   ;TO EQUAL 21501
020034  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
020036  005337  001202                          DEC     REPT            ;DONE YET
020042  001365                                  BNE     MREX2           ;NO LOOP
```

```
                                        ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
                                                MRCLK                   ;CLOCK MR REG
                                                MRCK                    ;CHECK MR REG
                                                171501                  ;TO EQUAL 171501
                                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                                MRCLK                   ;CLOCK MR REG
                                                MRCK                    ;MR REG SHOULD
                                                25501                   ;EQUAL 25501
                                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                                MRCLK
                                                MRCK
                                                175501
                                                HLT
                                        ;PERFORM NEXT STEP 3 TIMES TO FINISH WRITTING PREAMBLE
                                                MOV     #3,REPT
                                        MREX3:  MRCLK                   ;CLOCK MR REG
                                                MRCK                    ;CHECK MR REG
                                                25501                   ;TO EQUAL 25501
                                                HLT                     ;MR=BAD GOOD=CORRECT ANS
                                                MRCLK                   ;CLOCK MR REG
                                                MRCK                    ;CHECK MR REG
                                                175501                  ;TO EQUAL 175501
                                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
                                                DEC     REPT            ;DONE YES?
                                                BNE     MREX3           ;NO LOOP BACK

                                        ;MOVE DATA WORD INTO RS04 SHIFT REGISTER

                                                MRCLK                   ;CLOCK MR REG
                                                MRCK                    ;CHECK MR REG
                                                27501                   ;TO EQUAL 27501
                                                HLT                     ;MR=BAD GOOD=CORRECT ANS
                                                MRCLK                   ;CLOCK MR REG
                                                MRCK                    ;MR REG SHOULD
                                                123501                  ;EQUAL 123501
                                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;ENCODE SYNC 1

                                                MRCLK                   ;CLOCK MR REG
                                                MRCK                    ;MR REG SHOULD NOW
                                                73501                   ;EQUAL 73501
                                                HLT                     ;MR=BAD GOOD=CORRECT ANS
                                                MOV     #INBUF,R5       ;GET STARTING ADDR FOR DATA BUFFER
                                                MOV     (R5),R4         ;GET DATA
```

```
4053  020166  012737  002167  001214          MOV     #1143.,WORK     ;DOING A 1 SECTOR TRANSFER 127 WORDS
4054                                                                  ;18 BITS PER WORD-CLOCK LOOPS
4055                                                                  ;TAKE CARE OF 2 BITS AT A TIME
4056                                                                  ;127 TIMES 9 EQUALS 1143 LOOPS
4057                                                                  ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
4058  020174  042737  000200  001166          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
4059  020202  052737  000100  001166          BIS     #BIT6,ONCEE     ;SET 1ST TRANSFER WORD FLAG
4060  020210  104432                  1$:     XBIT                    ;GET 2 BITS OF DATA
4061  020212  104434                          CLKD1                   ;SEND FIRST CLOCK PULSE
4062                                                                  ;AND CALCULATE MR REG
4063                                                                  ;FOR CORRECT DATA (MWDT+MWDB)
4064  020214  104000                          HLT                     ;MR REG NOT CORRECT
4065  020216  104436                          CLKD2                   ;SEND 2ND CLOCK PULSE TO
4066                                                                  ;COMPLETE TRANSFER OF 2 BITS
4067                                                                  ;CALCULATE CORRECT ANS FOR
4068                                                                  ;MR REG (MWDT+MWDB)
4069  020220  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
4070  020222  032737  000200  001166          BIT     #BIT7,ONCEE     ;ON LAST WORD YET
4071  020230  001015                          BNE     2$              ;YES
4072  020232  032737  000400  001166          BIT     #BIT8,ONCEE     ;ON CRC WORD YET?
4073  020240  001043                          BNE     3$              ;YES
4074  020242  005337  001214                  DEC     WORK            ;DONE WITH 127 WORDS?
4075  020246  001360                          BNE     1$              ;NO
4076
4077  020250  052737  000200  001166          BIS     #BIT7,ONCEE     ;SET LAST WORD FLAG
4078  020256  012737  000012  001214          MOV     #10.,WORK       ;SET UP TO TRANSFER LAST WORD
4079  020264  005337  001214          2$:     DEC     WORK            ;DONE YET
4080  020270  001347                          BNE     1$
4081
4082  020272  052737  000400  001166          BIS     #BIT8,ONCEE     ;SET TRANSFERRING CRC WORD
4083  020300  042737  000200  001166          BIC     #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
4084
4085                          ;GENERATE RMR ERROR BY ATTEMPTING TO WRITE RSER
4086                          ;EXC SHOULD THEN BE ASSERTED
4087
4088  020306  012777  177777  160600          MOV     #-1,@RSER
4089  020314  004737  024112                  JSR     PC,GENCRC       ;GENERATE CRC WORD
4090                                                                  ;AND LEAVE IN "WORK"
4091  020320  012702  027566                  MOV     #INBUF,R2       ;GO TO END
4092  020324  062702  000400                  ADD     #400,R2         ;OF DATA BUFFER
4093  020330  013712  001214                  MOV     WORK,@R2        ;LOAD CRC WORD
4094  020334  010205                          MOV     R2,R5           ;RESET POINTER FOR
4095  020336  162705  000002                  SUB     #2,R5           ;R5 FOR CRC WD
4096  020342  012737  000012  001214          MOV     #10.,WORK       ;SETUP TO XFER CRC
4097  020350  005337  001214          3$:     DEC     WORK            ;DONE YET?
4098  020354  001315                          BNE     1$              ;NO
```

```
                                        ;EBL SHOULD NOW ASSERT AND CRC BE WRITTEN
020356  104422                                MRCLK                   ;CLOCK MR REG TO STEP THROUGH DEAD BAND AREA
020360  104420                                MRCK                    ;CHECK MR REG
020362  153501                                153501                  ;TO EQUAL 103501
020364  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS

                                        ;LOOP 6 TIMES
020366  012737  000006  001202                MOV     #6,REPT
020374  104422                        4$:     MRCLK                   ;CLOCK MR REG
020376  104420                                MRCK                    ;CHECK MR REG
020400  003501                                3501                    ;TO EQUAL 53501
020402  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
020404  104422                                MRCLK                   ;CLOCK MR REG
020406  104420                                MRCK                    ;CHECK MR REG
020410  153501                                153501                  ;TO EQUAL 103501
020412  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
020414  005337  001202                        DEC     REPT            ;DONE LOOPING YET?
020420  001365                                BNE     4$              ;NO

                                        ;FINISH UP
020422  104422                                MRCLK                   ;CLOCK MR REG
020424  104420                                MRCK                    ;CHECK MR REG
020426  003501                                3501                    ;TO EQUAL 3501
020430  104000                                HLT                     ;MR REG=BAD GOOD=CORRECT ANS
020432  104422                                MRCLK                   ;CLOCK MR REG
020434  104420                                MRCK                    ;CHECK MR REG
020436  151501                                151501                  ;TO EQUAL 151501
020440  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;TRANSFER SHOULD NOW BE COMPLETE
020442  104422                                MRCLK                   ;CLOCK MR REG
020444  104420                                MRCK                    ;CHECK MRE REG
020446  002701                                2701                    ;TO EQUAL 2701
020450  104000                                HLT                     ;MR=BAD GOOD=CORRECT ANS
020452  000240                                NOP                     ;STALL FOR TIME
020454  104050                                HLT     !WC!DS          ;SHOULD NEVER GET HERE
020456  000424                                BR      INTMR1          ;BECAUSE DRIVE SHOULD HAVE INTERRUPTED.
                                                                        ;CAUSING JUMP TO INTMR.
                                                                        ;CHECK FOR ASSERTION OF FTS ATTN L
                                        ;NOW TEST CONTROLLER

020460  022777  144260  160412    INTMR:  CMP     #144260,@RSCS1  ;IS CS1 CORRECT?
020466  001401                            BEQ     .+4             ;YES
020470  104014                            HLT     !DA!WC          ;YES
020472  022777  177702  160404    5$:     CMP     #177702,@RSWC   ;IS WC REG CORRECT?
020500  001401                            BEQ     .+4             ;YES
020502  104010                            HLT     !WC             ;WC SHOULD BE = TO 177600
020504  022777  000004  160402            CMP     #4,@RSER        ;DID RMR SET IN RSER
020512  001401                            BEQ     .+4             ;YES
020514  104050                            HLT     !DS!WC          ;RSER SHOULD = 4
020516  022777  000001  160364            CMP     #1,@RSDA        ;DOES RSDA=1
020524  001401                            BEQ     .+4             ;YES
020526  104004                            HLT     !DA             ;RSDA SHOULD=1
020530  000240                    INTMR1: NOP                     ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 66                    DISK ADDRESS OVERFLOW TEST
                                ;*****************************************************************************
020532  104400                  TST66:  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.

020534  104414                  MRAOE:  CLRDK                   ;CLEAR ALL REGISTERS
020536  012706  000500                  MOV     #500,SP         ;SETUP STACK POINTER
020542  104430                          MRIND                   ;SEND INDEX PULSE TO MR REG
020544  104420                          MRCK                    ;CHECK MAINT REG
020546  022701                          22701                   ;TO EQUAL 22701
020550  104424                          MRINT                   ;INITIALIZE MAINT REG BY SENDING
                                                                ;2 CLOCK PULSES (CLEAR MRSP)
020552  012777  007777  160330          MOV     #7777,@RSDA     ;SETUP DISK ADDRESS
020560  012777  177400  160316          MOV     #-400,@RSWC     ;SETUP FOR A 2 SECTOR TRANSFER
020566  012777  030366  160312          MOV     #OUTBUF,@RSBA   ;GET OUTPUT BUFFER

                                ;SETUP BUFFER WITH ALL ONES
020574  012705  030366                  MOV     #OUTBUF,R5      ;GET STARTING ADDRESS OF OUTBUF
020600  012737  000400  001202          MOV     #400,REPT       ;LOAD 2 SECTORS
020606  012725  177777          1$:     MOV     #-1,(R5)+       ;WITH WORDS
020612  005337  001202                  DEC     REPT            ;OF ALL ONES
020616  001373                          BNE     1$

020620  012777  000061  160252          MOV     #61,@RSCS1      ;LOAD WRITE COMMAND
020626  104430                          MRIND                   ;SET INDEX PULSE

                                ;SUPPLY CLOCKS TO STEP THROUGH A TRACK

020630  012737  000003  001202          MOV     #3,REPT
020636  012704  160000          5$:     MOV     #57344.,R4      ;SETUP FOR FAST CLOCK PULSES 172032 CLOCKS
020642  012702  000011                  MOV     #11,R2          ;(3 X 57344 = 172032)
020646  012703  000001                  MOV     #1,R3
020652  010277  160246          2$:     MOV     R2,@RSMR
020656  010377  160242                  MOV     R3,@RSMR
020662  005304                          DEC     R4
020664  001372                          BNE     2$
020666  005337  001202                  DEC     REPT
020672  001361                          BNE     5$

                                ;CAUSE "LBT IN RSDS TO SET

020674  104422                          MRCLK                   ;CLOCK AN 11 AND A 1 INTO RSMR
020676  104426                          DSCK                    ;CHECK MR
020700  012400                          12400                   ;TO EQUAL 12400
020702  104000                          HLT                     ;LBT SHOULD BE SET IN RSDS
```

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

020704  104430                          MRIND                   ;ASSERT MAINT INDEX PULSE
020706  005037  001174                  CLR     MCCNT           ;CLEAR THE CLOCK COUNTER
020712  104420                          MRCK                    ;CHECK MR REG
020714  002501                          2501                    ;TO EQUAL 2501. SHOULD STILL BE WRITING
020716  104000                          HLT

                                        ;SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS04 RESYNC PERIOD
020720  012737  001000  001202          MOV     #512.,REPT      ;CLOCK COUNT TO STEP THRU RESYNC
020726  104422                  4$:     MRCLK                   ;2ND REVOLUTION
020730  104420                          MRCK                    ;CHECK MR
020732  052501                          52501                   ;TO EQUAL 52501
020734  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
020736  104422                          MRCLK                   ;CLOCK MR REG
020740  104420                          MRCK                    ;CHECK MR
020742  002501                          2501                    ;REG TO
020744  104000                          HLT                     ;EQUAL 2501
020746  005337  001202                  DEC     REPT
020752  001365                          BNE     4$              ;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

020754  104422                  AOECK:  MRCLK
020756  104422                          MRCLK                   ;CAUSE SECTOR PULSE AND AOE ERROR
020760  104420                          MRCK                    ;CHECK FOR SECTOR PULSE
020762  022301                          22301                   ;IN RSMR
020764  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
020766  022777  001000  160120          CMP     #1000,@RSER     ;DID AOE SET IN RSER?
020774  001401                          BEQ     1$              ;AOE SHOULD BE SET IN RSER
020776  104040                          HLT     !DS             ;RSER SHOULD EQUAL 1000
021000  022777  152600  160104  1$:     CMP     #152600,@RSDS   ;IS RSDS CORRECT
021006  001401                          BEQ     2$              ;YES
021010  104040                          HLT     !DS             ;ERR & ATA SHOULD BE SET IN RSDS
                                                                ;BECAUSE OF AOE ERROR IN RSER
021012  104414                  2$:     CLRDK                   ;CLEAR ERROR
021014  005777  160074                  TST     @RSER           ;DID ERROR CLEAR?
021020  001401                          BEQ     3$              ;YES
021022  104040                          HLT     !DS             ;AOE DID NOT CLEAR BY SETTING CLR IN RSCS2
021024  022777  010600  160060  3$:     CMP     #10600,@RSDS    ;DID ERRORS CLEAR
021032  001401                          BEQ     .+4             ;YES
021034  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 INTO 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 67                 MAINTENANCE MODE VERIFY TEST
                                ;*************************************************************************
021036  104400                  TST67:  SCOPE

                                ;MODULE TESTED G182

021040  032777  004000  157760  MRVR:   BIT     #BIT11,@SWR     ;DO THIS TEST?
021046  001002                          BNE     3$              ;YES
021050  000137  021564                  JMP     @#INFTST        ;NO
021054  005037  001144          3$:     CLR     FLAG2           ;SET VERIFY TEST FLAG
021060  104414                          CLRDK                   ;CLEAR ALL DRIVES
021062  012737  007777  001230          MOV     #7777,WORK5     ;STALL TO
021070  005337  001230          4$:     DEC     WORK5           ;RESYNC DRIVE
021074  001375                          BNE     4$              ;TIMING LOGIC
021076  042737  000040  001166          BIC     #BIT5,ONCEE     ;CLEAR CLK CNT
021104  012777  160000  157772          MOV     #-20000,@RSWC   ;WRITE ONE TRACK - 8K WDS
021112  012737  177777  027566          MOV     #177777,INBUF   ;WRITE A PATTERN 12525
021120  052777  000010  157754          BIS     #BIT3,@RSCS2    ;SET BAI BIT
021126  012777  027566  157752          MOV     #INBUF,@RSBA    ;SET DATA WD
021134  012737  007777  001202          MOV     #7777,REPT      ;SETUP WAIT LOOP
021142  012777  000061  157730          MOV     #61,@RSCS1      ;GO WRITE
021150  105777  157724          1$:     TSTB    @RSCS1          ;DONE YET?
021154  100404                          BMI     2$              ;YES
021156  005337  001202                  DEC     REPT            ;DECREMENT COUNTER WAITING
021162  001372                          BNE     1$              ;FOR READY
021164  104000                          HLT                     ;READY NEVER CAME UP
021166  005777  157706          2$:     TST     @RSCS1          ;ANY ERRORS?
021172  100002                          BPL     MRVR1           ;NO
021174  104050                          HLT     !DS!WC          ;STOP HERE TILL THIS PROBLEM IS FIXED TRY DZRSB DIAG
021176  000433                          BR      TBDIA           ;TYPE MESSAGE
```

```
 4292  021200  104414                    MRVR1:  CLRDK                   ;CLEAR ALL REGISTERS
 4293  021202  012777  160000  157674            MOV     #-20000,@RSWC   ;SETUP WC
 4294  021210  052777  000010  157664            BIS     #BIT3,@RSCS2    ;SET BAI
 4295  021216  012777  027566  157662            MOV     #INBUF,@RSBA    ;SETUP RSBA
 4296  021224  012737  007777  001202            MOV     #7777,REPT      ;SETUP WAIT LOOP
 4297  021232  012777  000051  157640            MOV     #51,@RSCS1      ;DO A WRITE CHECK TO VERIFY DISK
 4298  021240  105777  157634            1$:     TSTB    @RSCS1          ;TEST
 4299  021244  100404                            BMI     2$              ;FOR READY TO COME BACK
 4300  021246  005337  001202                    DEC     REPT            ;WAIT
 4301  021252  001372                            BNE     1$              ;
 4302  021254  104000                            HLT                     ;READY NEVER CAME BACK
 4303  021256  005777  157616            2$:     TST     @RSCS1          ;ANY ERRORS?
 4304  021262  100032                            BPL     MRVRR           ;NO
 4305  021264  104050                            HLT     !DS!WC          ;STOP HERE WC FAILED
 4306                                                                    ;GO TO DZRSB DIAG
 4307                                                                    ;BEFORE TRYING TO DEBUG THLIS TEST
 4308  021266                            TBDIA:
 4309  021266  104402  021272                    TYPE    .+2             ;.ASCIZ <15><12>"FAILED VERIFY TEST --- RUN DZRSB DIAGNO
 4310  021350  000137  012060            MRVRR:  JMP     @#MRWRT         ;GO WRITE IN MAINTENANCE MODE
 4311                                    ;NOW CHECK TO SEE IF DRIVE WAS WRITTEN ON IN MAINTENANCE MODE
 4312
 4313  021354  104414                    MRVR2:  CLRDK                   ;CLEAR ALL REGISTERS
 4314  021356  012737  007777  001214            MOV     #7777,WORK      ;STALL - TO RESPONSE
 4315  021364  005337  001214            3$:     DEC     WORK            ;INDEX PULSE
 4316  021370  001375                            BNE     3$              ;ON DRIVE
 4317  021372  012777  160000  157504            MOV     #-20000,@RSWC   ;SETUP WC FOR 1 TRACK
 4318  021400  052777  000010  157474            BIS     #BAI,@RSCS2     ;SET BAI
 4319  021406  012777  027566  157472            MOV     #INBUF,@RSBA    ;SETUP RSBA
 4320  021414  012737  177777  027566            MOV     #177777,INBUF   ;SETUP FOR COMPARE
 4321  021422  012777  000051  157450            MOV     #51,@RSCS1      ;DO A WRITE CHECK
 4322  021430  105777  157444            1$:     TSTB    @RSCS1          ;TEST FOR
 4323  021434  100375                            BPL     1$              ;READY TO COME BACK
 4324  021436  032777  040000  157436            BIT     #WCE,@RSCS2     ;DID WCE SET?
 4325  021444  001442                            BEQ     2$              ;NO
 4326  021446  104402  021452                    TYPE    .+2             ;.ASCIZ <15><12> "WRITE AMPLIFIER DID NOT GET DISABLED B
 4327  021546  104040                            HLT     !DS
 4328  021550  000404                            BR      4$              ;GET OUT
 4329  021552  005777  157322            2$:     TST     @RSCS1          ;ANY ERRORS?
 4330  021556  100001                            BPL     4$              ;NO
 4331  021560  104040                            HLT     !DS             ;SHOULD NOT BE ANY ERRORS
 4332                                                                    ;TRY THE DZRSB DIAGNOSTIC
 4333  021562  000240            4$:     NOP
 4334
 4335  021564  000137  002120    INFTST: JMP     @#TRYNX                 ;GET NEXT DRIVE
```

```
                                        .SBTTL          $DONE - BELL AND SCOPE ROUTINE
                                        
021570  104400                  DONE:   SCOPE                           ;TERMINATIONG SCOPE FOR LOOPING
021572  062737  000001  001006          ADD     #1,PCNT+2               ;ADD 1 TO THE PASS COUNT
021600  005537  001004                  ADC     PCNT                    ;MAKE IT DOUBLE PREC.
021604  032777  002000  157214          BIT     #SW10,@SWR              ;RING THE BELL?
021612  001004                          BNE     4$                      ;NO!
021614  104402  021620                  TYPE    .+2                     ;.ASCIZ <BELL><177>
021624  013700  000042          4$:     MOV     @#42,R0                 ;GET MONITOR ADDRESS
021630  001405                          BEQ     SEND1                   ;IF NONE
021632  000005                          RESET
021634  004710                  SENDAD: JSR     7,(0)                   ;GO TO MONITOR
021636  000240  000240  000240          240,240,240                     ;SAVE ROOM FOR ACT11
021644  000137  021652          SEND1:  JMP     MULSYS                  ;RETURN
                                        
021650  000000                  .TBIT:  0                               ;T BIT FLAG
                                        
                                        ;MULTI DRIVE SYSTEM?
                                        
021652                          MULSYS:
021652  104402  021656                  TYPE    ,.+2                    ;.ASCIZ <15><12>"END OF PASS"
021674  005037  001010                  CLR     LAD
021700  005037  001000                  CLR     ICNT
021704  032737  000020  001166          BIT     #BIT4,ONCEE             ;MULTI DRVIE?
021712  001002                          BNE     1$                      ;NO
021714  000137  001674                  JMP     @#MULTII                ;YES
021720  000137  002324          1$:     JMP     @#NOWGO                 ;TEST ONLY ONE DRIVE
                                        
                                        ;ERROR TYPEOUT ROUTINE FOR NO-OP TEST
                                        
021724  032737  000004  001166  NOPERR: BIT     #BIT2,ONCEE             ;WERE WE HERE BEFORE?
021732  001031                          BNE     1$                      ;YES
021734  052737  000004  001166          BIS     #BIT2,ONCEE             ;SET BEEN HERE BEFORE FLAG
021742  104402  021746                  TYPE    .+2                     ;.ASCIZ <15><12>"ERROR CAUSED BY NO-OP FUNCTION "
022010  013746  001214                  MOV     WORK,-(6)               ;PUT WORK ON STACK
022014  104406                          TYPES                           ;TYPE STACK IN OCTAL - SUPRESS
022016  000207                  1$:     RTS     PC
```

```
4373  022020  104402  022034          CHG:    TYPE    ,REGCHG         ;TYPE MESSAGE
4374  022024  013746  001214                  MOV     WORK,-(6)       ;PUT WORK ON STACK
4375  022030  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
4376  022032  000207                          RTS     PC
4377
4378  022034  044103  047101  042507  REGCHG: .ASCIZ  "CHANGED WITH NO-OP FUNCTION "
4379  022042  020104  044527  044124
4380  022050  047040  026517  050117
4381  022056  043040  047125  052103
4382  022064  047511  020116     000
4383
4384  022071     015  051012  051115  TRMR:   .ASCIZ  <15><12>"RMR  DID NOT SET BY WRITING INTO "
4385  022076  020040  044504  020104
4386  022104  047516  020124  042523
4387  022112  020124  054502  053440
4388  022120  044522  044524  043516
4389  022126  044440  052116  020117
4390  022134     000
4391          022136                  .EVEN
4392
4393  022136  104422                  .MRINT: MRCLK                   ;CLOCK THE MAINT REG WITH AN 11 AND A 1
4394  022140  104422                          MRCLK                   ;SAME
4395  022142  000002                          RTI                     ;RETURN
4396
4397  022144  012777  000011  156752  .MRCLK: MOV     #11,@RSMR       ;CLOCK THE
4398  022152  012777  000001  156744          MOV     #1,@RSMR        ;MAINT REG
4399  022160  062737  000001  001176          ADD     #1,MCCNT+2      ;ADD 1 TO CLOCK COUNT
4400  022166  005537  001174                  ADC     MCCNT           ;MAKE DOUBLE PRECISION
4401  022172  000002                          RTI
4402
4403  022174  017700  156724          .MRCK:  MOV     @RSMR,BAD       ;GET THE CONTENTS OF RSMR
4404  022200  017601  000000                  MOV     @(SP),GOOD      ;GET THE CORRECT ANSWER
4405  022204  062716  000002                  ADD     #2,(SP)         ;UPDATE THE RETURN ADDRESS FOR AN ERROR
4406  022210  020100                          CMP     GOOD,BAD        ;IS THE MR REG CORRECT?
4407  022212  001002                          BNE     1$              ;NO EXIT
4408  022214  062716  000002                  ADD     #2,(SP)         ;UPDATE RETURN ADDRESS TO SKIP THE HLT FOR CORRECT ANS
4409  022220  000002                  1$:     RTI                     ;RETURN
4410
4411                                  ;SEND INDEX PULSE TO THE MAINTENANCE REGISTER
4412  022222  012777  000021  156674  .MRIND: MOV     #21,@RSMR       ;SEND INDEX
4413  022230  012777  000001  156666          MOV     #1,@RSMR        ;PULSE TO MR REG
4414  022236  000002                          RTI
4415  022240  017700  156646          .DSCK:  MOV     @RSDS,BAD       ;GET THE CONTENTS OF RSDS
4416  022244  017601  000000                  MOV     @(SP),GOOD      ;GET THE CORRECT ANS
4417  022250  062716  000002                  ADD     #2,(SP)         ;UPDATE THE RETURN ADDR FOR AN ERROR
4418  022254  020100                          CMP     GOOD,BAD        ;IS RSDS CORRECT
4419  022256  001002                          BNE     1$              ;NO EXIT
4420  022260  062716  000002                  ADD     #2,(SP)         ;UPDATE RETURN ADDR TO SKIP THE HLT FOR CORRRECT ANS
4421  022264  000002                  1$:     RTI
```

```
                                        ;GET 2 BITS OF DATA FROM BUFFER
                                        ;SET NOWOD (ODD DATA BIT NOW BEING WRITTEN) TO A 1 IF DATA TRANSFER BIT IS A 1
                                        ;CLEAR NOWOD IF DATA BIT IS A 0
                                        ;SET NOWEV (EVEN DATA BIT NOW BEING WRITTEN) TO A 1 IF DATA TRANSFER BIT IS A 1
                                        ;CLEAR NOWEV IF DATA BIT IS A 0

022266  032737  000100  001166  .XBIT:  BIT     #BIT6,ONCEE     ;1ST 2 BITS OF 1ST WORD?
022274  001427                          BEQ     2$              ;NO
022276  012737  000001  001146          MOV     #1,LSTEV        ;SET LAST EVEN BIT TRANSFERRED TO A 1
022304  012737  000001  001150          MOV     #1,LSTOD        ;SET LAST ODD BIT TRANFERRED TO A 1
                                                                ;THIS SETS UP THE SYNC 1 BITS AT END OF PREAMBLE
                                                                ;FOR THE TOP AND BOTTOM
                                                                ;BITS IN THE MR REGISTER
022312  042737  000100  001166          BIC     #BIT6,ONCEE     ;CLEAR 1ST WORD TRANSFER FLAG
022320  005037  001206          4$:     CLR     CLKCNT          ;CLEAR CLOCK COUNTER AT START OF EACH WORD
022324  032737  000400  001166          BIT     #BIT8,ONCEE     ;CRC WORD BEING WRITTEN?
022332  001042                          BNE     1$              ;YES
022334  005037  001154                  CLR     NOWOD           ;NO, LOAD EVEN
022340  005037  001152                  CLR     NOWEV           ;AND ODD WITH 0 FOR BITS 16 & 17 IN RS04 DATA WORD.
022344  012737  000010  001224  6$:     MOV     #8.,WORK3       ;8 LOOPS FOR REMAINING 16 BITS OF WORD
022352  000002                          RTI
022354  013737  001154  001150  2$:     MOV     NOWOD,LSTOD
022362  013737  001152  001146          MOV     NOWEV,LSTEV     ;SAVE LAST 2 BITS TRANSFERRED
022370  005737  001224                  TST     WORK3           ;DONE WITH WORD YET?
022374  001004                          BNE     3$              ;NO
022376  062705  000002                  ADD     #2,R5           ;UPDATE BUFFER WD
022402  011504                          MOV     (R5),R4         ;GET DATA WD
022404  000745                          BR      4$              ;GET BITS 16 & 17
022406  005037  001154          3$:     CLR     NOWOD           ;CLEAR PRESENT ODD BIT
022412  006104                          ROL     R4              ;GET NEXT ODD DATA BIT
022414  006137  001154                  ROL     NOWOD           ;SAVE IT IN ODD BIT
022420  005037  001152                  CLR     NOWEV           ;CLEAR PRESENT EVEN BIT
022424  006104                          ROL     R4              ;GET NEXT EVEN BIT
022426  006137  001152                  ROL     NOWEV           ;SAVE IT IN EVEN BIT
022432  005337  001224                  DEC     WORK3           ;KEEP COUNT OF BITS IN THE WORD
022436  000002                          RTI                     ;RETURN

                                        ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS. 0 & 1 ARE ALWAYS 0
022440  005037  001154          1$:     CLR     NOWOD           ;GET BITS 17
022444  006104                          ROL     R4              ;AND 16
022446  006137  001154                  ROL     NOWOD           ;FOR CRC WORD
022452  005037  001152                  CLR     NOWEV
022456  006104                          ROL     R4
022460  006137  001152                  ROL     NOWEV
022464  000727                          BR      6$              ;CONTINUE
```

```
4468                                    ;CLOCK ROUTINE (1ST OF TWO) WHICH IS USED TO CLOCK TWO BITS OF
4469                                    ;DATA TO THE DRIVE AT A TIME.  THIS ROUTINE ALSO CHECKS THE PREVIOUS
4470                                    ;BITS THAT HAVE BEEN TRANSFERRED AND CALCULATES WHICH STATE
4471                                    ;THE MWDT BIT (BIT 14 IN THE MR REG) AND MWDB BIT (BIT 12 IN THE MR REG) SHOULD BE IN
4472
4473
4474  022466  104422                    .CLKD1: MRCLK                   ;CLOCK MR REG WITH AN 11 AND A 1
4475  022470  005003                            CLR     R3              ;CLEAR WORK LOCATION
4476  022472  005737  001154                    TST     NOWOD           ;TEST ODD BIT NOW BEING SENT FOR A 1 OR A 0
4477  022476  001005                            BNE     TSTEVB          ;NOW TEST EVEN DATA BIT ON 1ST CLOCK
4478                                                                    ;NOW BIT IS A 1 MWDB IS 0
4479  022500  005737  001150            1$:     TST     LSTOD           ;TEST THE LAST ODD DATA BIT THAT WAS SENT
4480  022504  001002                            BNE     TSTEVB          ;LAST ODD DATA BIT WAS A 1
4481                                                                    ;MWDB IS A 0
4482
4483  022506  052703  010000            2$:     BIS     #BIT12,R3       ;SET MWDB FOR LATER COMPARE WITH MR REG
4484
4485                                    ;NOW TEST FOR EVEN BITS BEING SENT
4486
4487  022512  005737  001152            TSTEVB: TST     NOWEV           ;TEST EVEN BIT NOW BEING TRANSFERRED
4488                                                                    ;FOR EITHER A 1 OR A 0
4489  022516  001005                            BNE     1$              ;NOW BIT IS A 1
4490  022520  005737  001146                    TST     LSTEV           ;WAS LAST EVEN DATA BIT A 0?
4491  022524  001002                            BNE     1$              ;NO LAST EVEN DATA BIT WAS A 1
4492  022526  052703  040000                    BIS     #BIT14,R3       ;MWDT SHOULD BE SET
4493  022532  012701  123501            1$:     MOV     #123501,GOOD    ;GET CORRECT ANS
4494  022536  050301                            BIS     R3,GOOD         ;FOR MR REG
4495  022540  004737  023744                    JSR     PC,MRCAL        ;DETERMENE STATE OF SB & LSR BITS
4496  022544  017700  156354                    MOV     @RSMR,BAD       ;GET CONTENTS OF MR REG
4497  022550  020100                            CMP     GOOD,BAD        ;IS MR REG CORRECT?
4498  022552  001002                            BNE     2$              ;NO TYPE OUT MR REG
4499  022554  062716  000002                    ADD     #2,(SP)         ;UPDATE RETURN ADDR FOR CORRECT ANS
4500  022560  000002                    2$:     RTI                     ;RETURN
```

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

022562  104422                  .CLKD2: MRCLK                   ;CLOCK MR REG
022564  005737  001154                  TST     NOWOD           ;IS THE PRESENT DATA BIT A 1?
022570  001403                          BEQ     1$              ;NO IT IS A 0
022572  052703  010000                  BIS     #BIT12,R3       ;SET MWDB FOR BIT BEING SENT IS A 1
022576  000402                          BR      2$
022600  042703  010000          1$:     BIC     #BIT12,R3       ;CLEAR MWDB FOR PRESENT BIT IS A 0
022604  005737  001152          2$:     TST     NOWEV           ;IS PRESENT EVEN BIT A 1
022610  001403                          BEQ     3$              ;NO IT IS A 0
022612  052703  040000                  BIS     #BIT14,R3       ;IT IS A 1 SET MWDT
022616  000402                          BR      4$
022620  042703  040000          3$:     BIC     #BIT14,R3       ;PRESENT BIT IS A 0 CLEAR MWDT
022624  012701  023501          4$:     MOV     #23501,GOOD     ;GET CORRECT ANS
022630  050301                          BIS     R3,GOOD         ;FOR MR REG
022632  004737  023744                  JSR     PC,MRCAL        ;DETERMINE STATE OF SB & LSR BITS
022636  017700  156262                  MOV     @RSMR,BAD       ;GET CONTENTS OF MR REG
022642  020100                          CMP     GOOD,BAD        ;IS MR REG CORRECT?
022644  001002                          BNE     5$              ;NO TYPEOUT ERROR
022646  062716  000002                  ADD     #2,(SP)         ;UPDATE RETURN ADDR FOR CORRECT ANS
022652  000002          5$:     RTI                             ;RETURN

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

022654  012737  022764  001214  CRCTYP: MOV     #CRCTAB,WORK    ;GET STARTING LOC OF IC TABLE
022662  012737  000001  001220          MOV     #1,WORK1        ;SETUP TO TEST FIRST CHIP
022670  033737  001220  001172  1$:     BIT     WORK1,SAVEE     ;WAS IT THIS BIT?
022676  001006                          BNE     2$              ;YES TYPE IT
022700  062737  000006  001214          ADD     #6,WORK         ;NO INDEX TABLE POINTER
022706  006137  001220                  ROL     WORK1           ;SETUP TO TEST NEXT CHIP
022712  000766                          BR      1$              ;NOW TES IT
022714  004777  156274          2$:     JSR     PC,@WORK        ;TYPE OUT CHIP
022720  104402  022724                  TYPE    ,.+2            ;.ASCIZ " IN THE CRC REG SHOULD BE SET"
022762  000207                          RTS     PC
```

```
4538                                    ;TABLE FOR CRC TEST 2 TYPEOUT ROUTINE
4539
4540  022764  104402  023124    CRCTAB: TYPE    @E302
4541  022770  000207                    RTS     PC
4542  022772  104402  023132            TYPE    @E305
4543  022776  000207                    RTS     PC
4544  023000  104402  023140            TYPE    @E307
4545  023004  000207                    RTS     PC
4546  023006  104402  023146            TYPE    @E3010
4547  023012  000207                    RTS     PC
4548  023014  104402  023155            TYPE    @E3012
4549  023020  000207                    RTS     PC
4550  023022  104402  023164            TYPE    @E3015
4551  023026  000207                    RTS     PC
4552  023030  104402  023173            TYPE    @E242
4553  023034  000207                    RTS     PC
4554  023036  104402  023201            TYPE    @E245
4555  023042  000207                    RTS     PC
4556  023044  104402  023207            TYPE    @E247
4557  023050  000207                    RTS     PC
4558  023052  104402  023215            TYPE    @E2410
4559  023056  000207                    RTS     PC
4560  023060  104402  023224            TYPE    @E2412
4561  023064  000207                    RTS     PC
4562  023066  104402  023233            TYPE    @E2415
4563  023072  000207                    RTS     PC
4564  023074  104402  023242            TYPE    @E192
4565  023100  000207                    RTS     PC
4566  023102  104402  023250            TYPE    @E197
4567  023106  000207                    RTS     PC
4568  023110  104402  023256            TYPE    @E1910
4569  023114  000207                    RTS     PC
4570  023116  104402  023265            TYPE    @E1915
4571  023122  000207                    RTS     PC
```

```
 4572  023124  031505  026460  000062  E302:   .ASCIZ  "E30-2"
 4573  023132  031505  026460  000065  E305:   .ASCIZ  "E30-5"
 4574  023140  031505  026460  000067  E307:   .ASCIZ  "E30-7"
 4575  023146  031505  026460  030061  E3010:  .ASCIZ  "E30-10"
 4576  023154      000
 4577  023155      105  030063  030455  E3012:  .ASCIZ  "E30-12"
 4578  023162  000062
 4579  023164  031505  026460  032461  E3015:  .ASCIZ  "E30-15"
 4580  023172      000
 4581  023173      105  032062  031055  E242:   .ASCIZ  "E24-2"
 4582  023200      000
 4583  023201      105  032062  032455  E245:   .ASCIZ  "E24-5"
 4584  023206      000
 4585  023207      105  032062  033455  E247:   .ASCIZ  "E24-7"
 4586  023214      000
 4587  023215      105  032062  030455  E2410:  .ASCIZ  "E24-10"
 4588  023222  000060
 4589  023224  031105  026464  031061  E2412:  .ASCIZ  "E24-12"
 4590  023232      000
 4591  023233      105  032062  030455  E2415:  .ASCIZ  "E24-15"
 4592  023240  000065
 4593  023242  030505  026471  000062  E192:   .ASCIZ  "E19-2"
 4594  023250  030505  026471  000067  E197:   .ASCIZ  "E19-7"
 4595  023256  030505  026471  030061  E1910:  .ASCIZ  "E19-10"
 4596  023264      000
 4597  023265      105  034461  030455  E1915:  .ASCIZ  "E19-15"
 4598  023272  000065
```

```
4599                                    ;GET TWO BITS OF DATA FROM INBUF
4600                                    ;FOR READING FROM DRIVE TO DETERMINE THE
4601                                    ;STATE OF MRDT AND MRDB IN THE MR REG.
4602
4603  023274  005737  001224    .RBIT:  TST     WORK3           ;STARTING NEW WD?
4604  023300  001031                    BNE     3$              ;NO
4605  023302  062705  000002            ADD     #2,R5           ;UPDATE BUFFER WD
4606  023306  011504                    MOV     (R5),R4         ;GET DATA WD
4607  023310  005037  001206            CLR     CLKCNT          ;CLEAR CLOCK COUNTER AT START OF EACH WD
4608  023314  032737  000400  001166    BIT     #BIT8,ONCEE     ;ON CRC WD?
4609  023322  001035                    BNE     1$              ;YES
4610  023324  032737  000040  001144    BIT     #BIT5,FLAG2     ;IN CRC TEST ????
4611  023332  001404                    BEQ     7$              ;NO
4612  023334  012737  000010  001224    MOV     #8.,WORK3
4613  023342  000410                    BR      3$
4614  023344  005037  001154    7$:     CLR     NOWOD           ;LOAD EVEN & ODD BITS
4615  023350  005037  001152            CLR     NOWEV           ;WITH 0 FOR BITS 16 & 17 IN RS04 DATA WORD
4616  023354  012737  000010  001224    MOV     #8.,WORK3       ;8 LOOPS FOR REMAINING 16 BITS OF WORD
4617  023362  000002                    RTI
4618  023364  005037  001154    3$:     CLR     NOWOD           ;CLEAR PRESENT ODD BIT
4619  023370  006104                    ROL     R4              ;GET NEXT ODD DATA BIT
4620  023372  006137  001154            ROL     NOWOD           ;SAVE IT IN ODD BIT
4621  023376  005037  001152            CLR     NOWEV           ;CLEAR PRESENT EVEN BIT
4622  023402  006104                    ROL     R4              ;GET NEXT EVEN BIT
4623  023404  006137  001152            ROL     NOWEV           ;SAVE IT IN EVEN BIT
4624  023410  005337  001224            DEC     WORK3           ;KEEP COUNT OF BITS IN THE WORD
4625  023414  000002                    RTI                     ;RETURN
4626                                    ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS, 0 & 1 ARE ALWAYS 0
4627  023416  005037  001154    1$:     CLR     NOWOD           ;GET BITS 17
4628  023422  006104                    ROL     R4              ;AND 16
4629  023424  006137  001154            ROL     NOWOD           ;FOR CRC WORD
4630  023430  005037  001152            CLR     NOWEV
4631  023434  006104                    ROL     R4
4632  023436  006137  001152            ROL     NOWEV
4633  023442  000744                    BR      6$              ;CONTINUE
```

```
 4634  023444  004737  023706          .CLKR1: JSR     PC,CALRTB       ;CALCULATE TOP AND BOTTOM BITS FOR MR REG
 4635  023450  012703  000011                  MOV     #11,R3          ;SETUP CLOCK BITS
 4636  023454  053703  001214          CLODK:  BIS     WORK,R3         ;SET TOP & BOTTOM BITS
 4637  023460  010377  155440                  MOV     R3,@RSMR        ;SEND
 4638  023464  042703  000010                  BIC     #BIT3,R3        ;CLOCK
 4639  023470  010377  155430                  MOV     R3,@RSMR        ;PULSE
 4640  023474  062737  000001  001176          ADD     #1,MCCNT+2      ;INCREMENT
 4641  023502  005537  001174                  ADC     MCCNT           ;CLOCK COUNT
 4642  023506  012701  023601                  MOV     #23601,GOOD     ;CALCULATE CORRECT ANS FOR MR REG
 4643  023512  032737  000004  001144          BIT     #BIT2,FLAG2     ;WRITE CK TEST?
 4644  023520  001402                          BEQ     7$              ;NO
 4645  023522  052701  000100                  BIS     #BIT6,GOOD      ;YES SET RD IN MR REG
 4646  023526  050301                  7$:     BIS     R3,GOOD
 4647  023530  042701  000010                  BIC     #BIT3,GOOD      ;CLEAR MCLK
 4648  023534  032737  000400  001166          BIT     #BIT8,ONCEE     ;ON CRC WD?
 4649  023542  001406                          BEQ     5$              ;NO
 4650  023544  022737  000011  001202          CMP     #11,REPT        ;SHOULD CRCW BE SET?
 4651  023552  001402                          BEQ     5$              ;YES
 4652  023554  042701  020000                  BIC     #20000,GOOD     ;CLEAR CRCW
 4653  023560  032737  000001  001144  5$:     BIT     #BIT0,FLAG2     ;SHOULD SDCLK BE SET?
 4654  023566  001004                          BNE     1$              ;YES
 4655  023570  052737  000001  001144          BIS     #BIT0,FLAG2     ;NO
 4656  023576  000405                          BR      2$              ;CONTINUE
 4657  023600  052701  100000          1$:     BIS     #BIT15,GOOD     ;SET IT
 4658  023604  042737  000001  001144          BIC     #BIT0,FLAG2     ;CLEAR FLAG FOR SDCLK FOR NEXT CLOCK PULSE
 4659  023612  005337  001204          2$:     DEC     REPT1           ;SHOULD SB SET?
 4660  023616  001017                          BNE     6$              ;NO
 4661  023620  012737  000022  001204          MOV     #18.,REPT1      ;RESET SB COUNTER
 4662  023626  052701  004000                  BIS     #BIT11,GOOD     ;SET SB
 4663  023632  032737  000400  001166  3$:     BIT     #BIT8,ONCEE     ;ON CRC WD?
 4664  023640  001406                          BEQ     6$              ;NO
 4665  023642  022737  000022  001204          CMP     #22,REPT1       ;SHOULD SB AND CRCW BE SET ?
 4666  023650  001002                          BNE     6$              ;NO
 4667  023652  052701  020000                  BIS     #20000,GOOD     ;SET SB AND CRCW
 4668  023656  017700  155242          6$:     MOV     @RSMR,BAD       ;GET MR REG
 4669  023662  020100                          CMP     GOOD,BAD        ;IS RSMR CORRECT?
 4670  023664  001002                          BNE     4$              ;NO
 4671  023666  062716  000002                  ADD     #2,(SP)         ;YES
 4672  023672  000002                  4$:     RTI                     ;RETURN
 4673
 4674  023674  004737  023706          .CLKR2: JSR     PC,CALRTB
 4675  023700  012703  050011                  MOV     #50011,R3
 4676  023704  000663                          BR      CLODK
```

```
4677                                    ;CALCULATE THE STATE OF MRDT AND MRDB FROM CURRENT INPUT BITS
4678                                    ;LOCATION WORK CONTAINS CORRECT DATA FOR MRDT AND MRDB
4679  023706  005037  001214            CALRTB: CLR     WORK            ;CLEAR WORK LOCATION
4680  023712  005737  001154                    TST     NOWOD           ;IS CURRENT ODD BIT A 0?
4681  023716  001403                            BEQ     1$              ;YES
4682  023720  052737  000004  001214            BIS     #BIT2,WORK      ;NO SET MRDB
4683  023726  005737  001152            1$:     TST     NOWEV           ;IS CURRENT EVEN BIT A 0?
4684  023732  001403                            BEQ     2$              ;YES
4685  023734  052737  000040  001214            BIS     #BIT5,WORK      ;NO, SET MRDT
4686  023742  000207                    2$:     RTS     PC              ;RETURN
4687
4688                                    ;CALCULATE MR REG TO DETERMINE THE STATE OF THE CRC-SB AND LSR BITS
4689                                    ;ON THE DIFFERENT CLOCKS ON THE DIFFERENT WORDS THROUGHOUT THE SECTOR
4690
4691  023744  005237  001206            MRCAL:  INC     CLKCNT          ;ADD ONE TO CLOCK COUNT OF WORD
4692  023750  032737  000200  001166            BIT     #BIT7,ONCEE     ;TRANSFERRING LAST WORD?
4693  023756  001026                            BNE     LSTWD           ;YES
4694  023760  032737  000400  001166            BIT     #BIT8,ONCEE     ;TRANSFERRING CRC WORD?
4695  023766  001040                            BNE     CRCWD           ;YES
4696  023770  022737  000010  001206            CMP     #8.,CLKCNT      ;CLOCK COUNT 8 OR GREATER?
4697  023776  101401                            BLOS    1$              ;YES
4698  024000  000414                            BR      2$              ;GET OUT
4699  024002  022737  000021  001206    1$:     CMP     #17.,CLKCNT     ;CLOCK COUNT 17 OR GREATER?
4700  024010  101410                            BLOS    2$              ;YES GET OUT
4701
4702  024012  052701  004000                    BIS     #BIT11,GOOD     ;SET SB BIT
4703  024016  022737  000017  001206            CMP     #15.,CLKCNT     ;SHOULD LSR BE CLEARED
4704  024024  001002                            BNE     2$              ;NO
4705  024026  042701  002000                    BIC     #BIT10,GOOD     ;CLEAR LSR
4706  024032  000207                    2$:     RTS     PC              ;RETURN
4707
4708                                    ;CALCULATE MR FOR LAST DATA WORD
4709  024034  022737  000016  001206    LSTWD:  CMP     #14.,CLKCNT     ;IS THIS CLOCK 14 OR LESS?
4710  024042  103011                            BHIS    2$              ;YES GETOUT
4711  024044  022737  000017  001206            CMP     #[illegible],CLKCNT ;IS THIS CLOCK 15?
4712  024052  001003                            BNE     [illegible]     ;NO
4713  024054  042701  002000                    BIC     #BIT10,GOOD     ;YES CLEAR LSR
4714  024060  000402                            BR      2$              ;GET OUT
4715  024062  042701  020000            1$:     BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
4716  024066  000207                    2$:     RTS     PC
4717
4718                                    ;CALCULATE MR FOR CRC WORD
4719
4720  024070  042701  020000            CRCWD:  BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
4721  024074  022737  000017  001206            CMP     #17,CLKCNT      ;IS THIS CLOCK 17?
4722  024102  001002                            BNE     1$              ;NO
4723  024104  042701  002000                    BIC     #BIT10,GOOD     ;CLEAR LSR BIT
4724  024110  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

024112  012737  000200  001202  GENCRC: MOV     #128.,REPT      ;128 WORDS PER SECTOR
024120  032737  000040  001144          BIT     #BIT5,FLAG2     ;IN CRC TEST?
024126  001403                        BEQ     13$             ;NO
024130  012737  000220  001202        MOV     #144.,REPT      ;YES
024136  012705  027566          13$:  MOV     #INBUF,R5       ;GET STARTING ADDR OF OUTPUT BUFFER
024142  011504                        MOV     (R5),R4         ;GET DATA WD
024144  005037  001216                CLR     WORK0           ;CLEAR WORK LOCATION

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

024150  012737  000022  001204  1$:     MOV     #18.,REPT1      ;GET 18 BITS PER WD
024156  032737  000040  001144          BIT     #BIT5,FLAG2     ;IN CRC TEST?
024164  001403                        BEQ     2$              ;NO
024166  012737  000020  001204        MOV     #16.,REPT1      ;YES
024174  013737  001216  001200  2$:     MOV     WORK0,WCRC      ;SAVE CURRENT CRC WD
024202  005037  001212                CLR     WK15            ;CLEAR BIT 15 FROM CRC AT T 1
024206  000241                        CLC                     ;CLEAR CARY
024210  006137  001216                ROL     WORK0           ;SHIFT CRC WD LEFT
024214  006137  001212                ROL     WK15            ;CONTAINS BIT 15 OF CRC
024220  032737  000040  001144        BIT     #BIT5,FLAG2     ;IN CRC TEST?
024226  001004                        BNE     12$             ;YES
024230  022737  000021  001204        CMP     #17.,REPT1      ;DONE BITS 16 AND 17 YET?
024236  101406                        BLOS    3$              ;NO
024240  005037  001210          12$:  CLR     INBIT           ;CLEAR WORK LOC
024244  006104                        ROL     R4              ;PUT DATA BIT FROM BUFFER
024246  006137  001210                ROL     INBIT           ;IN WORK1 LOC
024252  000402                        BR      4$
024254  005037  001210          3$:   CLR     INBIT           ;FOR BITS 16 AND 17
024260  013737  001212  001214  4$:   MOV     WK15,WORK       ;GET BIT 15 OF CRC
024266  004737  024512          5$:   JSR     PC,XXOR         ;XOR BIT15 WITH INPUT BIT
024272  042737  000001  001216        BIC     #BIT0,WORK0
024300  005737  001210                TST     INBIT           ;TEST RESULT OF XOR
024304  001403                        BEQ     6$
024306  052737  000001  001216        BIS     #BIT0,WORK0
024314  013737  001210  001156  6$:   MOV     INBIT,RS0       ;SAVE XOR SESULT OF BIT 0 AND INPUT
```

```
                                        ;FROM B0 IN WORK0 AND B1 IN SAVED CRC (WCRC) CALCULATE
                                        ;NEW B2 FOR WORK0
                                
024322  005037  001214                  CLR     WORK
024326  032737  000002  001200          BIT     #BIT1,WCRC
024334  001403                          BEQ     7$
024336  052737  000001  001214          BIS     #BIT0,WORK
024344  013737  001156  001210  7$:     MOV     RS0,INBIT
024352  004737  024512                  JSR     PC,XXOR
024356  042737  000004  001216          BIC     #BIT2,WORK0
024364  005737  001210                  TST     INBIT           ;TEST RESULT OF XOR
024370  001403                          BEQ     8$
024372  052737  000004  001216          BIS     #BIT2,WORK0
                                
                                        ;FROM B0 IN WORK0 AND B14 IN WCRC CLACULATE BIT15 IN WORK0
                                
024400  005037  001214          8$:     CLR     WORK
024404  032737  040000  001200          BIT     #BIT14,WCRC
024412  001403                          BEQ     9$
024414  052737  000001  001214          BIS     #BIT0,WORK
024422  013737  001156  001210  9$:     MOV     RS0,INBIT
024430  004737  024512                  JSR     PC,XXOR
024434  042737  100000  001216          BIC     #BIT15,WORK0
024442  005737  001210                  TST     INBIT           ;TEST RESULT OF XOR
024446  001403                          BEQ     10$
024450  052737  100000  001216          BIS     #BIT15,WORK0
024456  005337  001204          10$:    DEC     REPT1           ;DONE WITH WD
024462  001244                          BNE     2$              ;NO
024464  005337  001202                  DEC     REPT            ;DONE WITH SECTOR?
024470  001404                          BEQ     11$             ;YES
024472  062705  000002                  ADD     #2,R5           ;GET NEXT WD
024476  011504                          MOV     (R5),R4         ;GET DATA WD
024500  000623                          BR      1$
024502  013737  001216  001214  11$:    MOV     WORK0,WORK      ;SAVE CRC WORD IN WORK
024510  000207                          RTS     PC              ;EXIT
                                
                                        ;XOR SUBROUTINE
                                
024512  013703  001214          XXOR:   MOV     WORK,R3
024516  043703  001210                  BIC     INBIT,R3
024522  043737  001214  001210          BIC     WORK,INBIT
024530  050337  001210                  BIS     R3,INBIT
024534  000207                          RTS     PC
```

```
4811                                    .SBTTL          $TYPE - TTY TYPEOUT ROUTINE
4812
4813                                    ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
4814                                    ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
4815                                    ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
4816                                    ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
4817                                    ;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
4818                                    ;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
4819                                    ;IS IN FILCHR+1.
4820
4821  024536  010446                    .TYPE:  MOV     R4,-(6)         ;SAVE R4
4822  024540  010546                            MOV     R5,-(6)         ;SAVE R5
4823  024542  017605  000004                    MOV     @4(6),R5        ;GET ADDRESS TO BE TYPED
4824  024546  032705  177400                    BIT     #177400,R5      ;IS IT A TYPEM?
4825  024552  001002                            BNE     1$              ;NO
4826  024554  016605  000004                    MOV     4(6),R5         ;GET ADDRESS OF CHARACTER
4827  024560  105715                    1$:     TSTB    (R5)            ;TERMINATOR?
4828  024562  001423                            BEQ     2$              ;GET OUT IF SO
4829  024564  122715  000012                    CMPB    #12,(R5)        ;IS THE CHAR A LINE FEED
4830  024570  001012                            BNE     4$              ;NO - GET OUT
4831  024572  113704  001015                    MOVB    FILCHR+1,R4     ;GET THE FILL COUNT
4832  024576  113777  001014  154220  5$:      MOVB    FILCHR,@TPB     ;TYPE A FILLER
4833  024604  105777  154206                    TSTB    @TPS            ;DONE YET?
4834  024610  100375                            BPL     .-4             ;NO - WAIT
4835  024612  005304                            DEC     R4              ;DEC COUNT
4836  024614  001370                            BNE     5$              ;LOOP UNTIL 0
4837  024616  112577  154202            4$:     MOVB    (R5)+,@TPB      ;LOAD AND TYPE THE CHARACTER
4838  024622  105777  154170                    TSTB    @TPS            ;IS THE PRINTER READY
4839  024626  100375                            BPL     .-4             ;WAIT UNTIL IT IS
4840  024630  000753                            BR      1$              ;GET THE NEXT CHARACTER
4841  024632  017646  000004            2$:     MOV     @4(6),-(6)      ;GET ADDRESS TO BE TYPED
4842  024636  062766  000002  000006          ADD     #2,6(6)         ;ADD 2 TO THE ADDRESS
4843  024644  022666  000004                    CMP     (6)+,4(6)       ;IS IT .+2?
4844  024650  001006                            BNE     3$              ;NO
4845  024652  062705  000002                    ADD     #2,R5           ;ADD 2 TO THE ADDRESS
4846  024656  042705  000001                    BIC     #1,R5           ;BACK UP TO AN EVEN BYTE
4847  024662  010566  000004                    MOV     R5,4(6)         ;RESTORE ADDRESS
4848  024666  012605                    3$:     MOV     (6)+,R5         ;RESTORE R5
4849  024670  012604                            MOV     (6)+,R4         ;RESTORE R4
4850  024672  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:"

024674  104452                  .SCOPE: KBDIN                   ;GO CHECK FOR ↑G
024676  032777  000400  154122          BIT     #SW8,@SWR       ;LOOP ON SPEC. TEST?
024704  001404                        BEQ     1$              ;NO LOOP ON SPEC. TEST
024706  127737  154114  001000        CMPB    @SWR,ICNT       ;ON RIGHT TEST?   *SW7-0*
024714  001453                        BEQ     .OVER           ;NOT RIGHT TEST
024716  032777  040000  154102  1$:   BIT     #SW14,@SWR      ;LOOP ON TEST?
024724  001045                        BNE     .KIT            ;LOOP ON TEST IS SET
024726  000416                        BR      3$              ;SKIP - NOP FOR XOR TESTER
024730  013746  000004                MOV     @#4,-(6)        ;PUSH @#4 ON STACK
024734  012737  024754  000004        MOV     #4$,@#4         ;SET FOR TIMEOUT
024742  005737  177060                TST     @#177060        ;ERROR ON XOR?
024746  012637  000004                MOV     (6)+,@#4        ;POP STACK INTO @#4
024752  000422                        BR      .SVLAD          ;NO ERROR - GO TO NEXT TEST
024754  022626                  4$:   CMP     (6)+,(6)+       ;CLEAR STACK
024756  012637  000004                MOV     (6)+,@#4        ;POP STACK INTO @#4
024762  000426                        BR      .KIT            ;ERROR - LOOP ON TEST
024764  032777  004000  154034  3$:   BIT     #SW11,@SWR      ;KILL ITERATIONS
024772  001012                        BNE     .SVLAD          ;YES - KILL ITERATIONS
024774  105737  001001                TSTB    ICNT+1          ;FIRST ONE?
025000  001404                        BEQ     2$              ;BRANCH IF FIRST
025002  123737  025066  001001        CMPB    TIMES,ICNT+1    ;DONE?
025010  003013                        BGT     .KIT            ;BRANCH IF NOT
025012  112737  000001  001001  2$:   MOVB    #1,ICNT+1       ;FIRST ITERATION
025020  105237  001000          .SVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
025024  011637  001010                MOV     (6),LAD         ;SAVE LOOP ADDRESS
025030  013777  001000  153772        MOV     ICNT,@DISPLAY   ;DISPLAY TEST NO. AND ITERATION COUNT
025036  000002                        RTI                     ;RETURN

025040  105237  001001          .KIT:   INCB    ICNT+1          ;INC THE ITERATION COUNT
025044  013777  001000  153756  .OVER:  MOV     ICNT,@DISPLAY   ;SET UP DISPLAY
025052  005737  001010                TST     LAD             ;FIRST ONE?
025056  001760                        BEQ     .SVLAD          ;YES
025060  013716  001010                MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
025064  000002                        RTI                     ;FIXES PS

025066  000001          TIMES:  1                       ;RUN 1 TIMES
```

```
4894                                            .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)
4895
4896                                            ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
4897                                            ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
4898                                            ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
4899                                            ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
4900                                            ;(HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADITIONAL TYPEOUTS.
4901
4902  025070  104452                    .HLT:   KBDIN                   ;GO CHECK FOR ↑G
4903  025072  032777  002000  153726            BIT     #SW10,@SWR      ;BELL ON ERROR?
4904  025100  001402                            BEQ     1$              ;NO - SKIP
4905  025102  104402  000007                    TYPE    ,BELL           ;RING BELL
4906  025106  005237  001002            1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
4907  025112  032777  020000  153706            BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
4908  025120  001025                            BNE     2$              ;SKIP TYPEOUTS
4909  025122  104402  025126                    TYPE    .+2             ;.ASCIZ <15><12>
4910  025132  011637  001012                    MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
4911  025136  162737  000002  001012            SUB     #2,HLTADR       ;FUDGE ADDRESS
4912  025144  117737  153642  025226            MOVB    @HLTADR,.HLTCT  ;GET HLT ARGUEMENT
4913  025152  013746  001012                    MOV     HLTADR,-(6)     ;PUT HLTADR ON STACK
4914  025156  104404                            TYPEO                   ;TYPE STACK IN OCTAL
4915  025160  104402  025164                    TYPE    .+2             ;.ASCIZ " "
4916  025170  004737  026374                    JSR     PC,RSREG        ;GO TO USER ERROR ROUTINE
4917  025174  005777  153626            2$:     TST     @SWR            ;HALT ON ERROR
4918  025200  100001                            BPL     .+4             ;SKIP IF CONTINUE
4919  025202  000000                            HALT                    ;HALT ON ERROR!
4920  025204  032777  001000  153614            BIT     #SW9,@SWR       ;CHECK FOR INHIBIT LOOP ON ERROR
4921  025212  001003                            BNE     3$              ;SKIP IF LOOP ON ERROR
4922  025214  105037  001001                    CLRB    ICNT+1          ;CLEAR ITERATION COUNT
4923  025220  000002                            RTI                     ;RETURN
4924  025222  000137  025040            3$:     JMP     .KIT            ;LOOP ON TEST UNTIL NO ERRORS
4925
4926  025226  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.

025230  012737  170101  025416  .TYPEB: MOV     #170101,.PR     ;SET BIT FLAG AND 16. CHARACTER COUNT
025236  000411                  BR      .PTIT           ;NOW TYPE IT IN BIT FORM
025240  112737  000001  025416  .TYPEO: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
025246  000402                  BR      .+6             ;SKIP
025250  005037  025416          .TYPES: CLR     .PR             ;SUPRESS LEADING ZERO'S
025254  112737  177772  025417          MOVB    #-6,.PR+1       ;SET COUNT
025262                          .PTIT:
025262  010446                  MOV     R4,-(6)         ;PUSH R4 ON STACK
025264  010546                  MOV     R5,-(6)         ;PUSH R5 ON STACK
025266  016605  000010          MOV     10(6),R5        ;GET THE DATA
025272  012704  025420          MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
025276  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
025300  000411                  BR      .PRF            ;ROTATE FIRST BIT
025302  105014          .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
025304  032737  000100  025416          BIT     #100,.PR        ;BIT TYPING MODE?
025312  001004                  BNE     .PRF            ;YES - SKIP 2 ROTATES
025314  006105                  ROL     R5              ;ROTATE BIT INTO C
025316  106114                  ROLB    (4)             ;PACK IT
025320  006105                  ROL     R5              ;ROTATE BIT INTO C
025322  106114                  ROLB    (4)             ;PACK IT
025324  006105          .PRF:   ROL     R5              ;ROTATE BIT INTO C
025326  106114                  ROLB    (4)             ;PACK IT
025330  105714                  TSTB    (4)             ;IS IT ZERO?
025332  001402                  BEQ     .+6             ;SKIP INC
025334  105237  025416          INCB    .PR             ;SET FILL SWITCH
025340  105737  025416          TSTB    .PR             ;CHECK FILL SWITCH
025344  001402                  BEQ     .+6             ;SKIP BITSET
025346  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
025352  105237  025417          INCB    .PR+1           ;INC COUNT
025356  001351                  BNE     .PRL            ;REPEAT
025360  022704  025420          CMP     #.PR+2,R4       ;EMPTY BUFFER?
025364  001002                  BNE     .+6             ;SKIP IF NOT
025366  112724  000060          MOVB    #'0,(4)+        ;LOAD 1 ZERO
025372  105014                  CLRB    (4)             ;NULL TERMINATOR
025374  104402  025420          TYPE    .PR+2           ;TYPE IT
025400  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
025402  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
025404  016666  000002  000004  MOV     2(6),4(6)       ;GET RID OF
025412  012616                  MOV     (6)+,(6)        ;DATA WORD
025414  000002                  RTI                     ;RETURN

025416  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'.

025442  012777  025570  000126  .POWER: MOV     #.ILLUP,@.PUVEC ;SET FOR FAST UP
025450  012777  000340  000122          MOV     #340,@.PUVEC+2  ;PRIO:7
025456  010046                        MOV     R0,-(6)         ;PUSH R0 ON STACK
025460  010146                        MOV     R1,-(6)         ;PUSH R1 ON STACK
025462  010246                        MOV     R2,-(6)         ;PUSH R2 ON STACK
025464  010346                        MOV     R3,-(6)         ;PUSH R3 ON STACK
025466  010446                        MOV     R4,-(6)         ;PUSH R4 ON STACK
025470  010546                        MOV     R5,-(6)         ;PUSH R5 ON STACK
025472  010637  025574                MOV     SP,.SAVR6       ;SAVE SP
025476  012777  025506  000072        MOV     #.POWUP,@.PUVEC ;SET UP VECTOR
025504  000000                        HALT                    ;WAIT FOR PF

025506  013706  025574  .POWUP: MOV     .SAVR6,SP       ;GET SP
025512  005001                  CLR     R1              ;WAIT LOOP FOR THE TTY
025514  005201          1$:     INC     R1              ;WAIT FOR THE INC
025516  001376                  BNE     1$              ;OF  WORD
025520  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
025522  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
025524  012603                  MOV     (6)+,R3         ;POP STACK INTO R3
025526  012602                  MOV     (6)+,R2         ;POP STACK INTO R2
025530  012601                  MOV     (6)+,R1         ;POP STACK INTO R1
025532  012600                  MOV     (6)+,R0         ;POP STACK INTO R0
025534  012737  025442  000024  MOV     #.POWER,@#24    ;SET UP THE POWER DOWN VECTOR
025542  012737  000340  000026  MOV     #340,@#26       ;PRIO:7
025550  104402  025554          TYPE    .+2             ;.ASCIZ <15><12>"POWER"
025564  000137  021652          JMP     MULSYS          ;JMP TO USER ADDRESS

025570  000000  .ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
025572  000776          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE

025574  000000  .SAVR6: 0                       ;PUT THE SP HERE
025576  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.

025602  011646                .RDOCT: MOV     (6),-(6)        ;MOVE THE PC
025604  016666  000004  000002          MOV     4(6),2(6)       ;MOVE THE PS
025612  010146                        MOV     R1,-(6)         ;PUSH R1 ON STACK
025614  010246                        MOV     R2,-(6)         ;PUSH R2 ON STACK
025616  010346                        MOV     R3,-(6)         ;PUSH R3 ON STACK
025620  104412                4$:     RDLIN                   ;READ A LINE INTO INPUT
025622  005001                        CLR     R1              ;INIT DATA WORD
025624  005037  027564                CLR     CTN             ;CLEAR COUNT WORD
025630  012703  026056                MOV     #INPUT,R3       ;INIT POINTER
025634  112302                1$:     MOVB    (3)+,R2         ;GET A BYTE
025636  122702  000015                CMPB    #15,R2          ;WAS IT A <CR>?
025642  001421                        BEQ     2$              ;GET OUT IF YES
025644  122702  000060                CMPB    #'0,R2          ;CHECK FOR 0 OR GREATER
025650  003024                        BGT     3$              ;ERROR - LESS THAN 0
025652  122702  000067                CMPB    #'7,R2          ;CHECK FOR 7 OR LESS
025656  002421                        BLT     3$              ;ERROR - GREATER THAN 7
025660  006002                        ROR     R2              ;GET
025662  006002                        ROR     R2              ;INTO
025664  006002                        ROR     R2              ;POSITION
025666  006101                        ROL     R1              ;FIRST BIT
025670  006102                        ROL     R2              ;GET
025672  006101                        ROL     R1              ;SECOND BIT
025674  006102                        ROL     R2              ;GET
025676  006101                        ROL     R1              ;THIRD BIT
025700  005237  027564                INC     CTN             ;YES HE TYPED SOMETHING
025704  000753                        BR      1$              ;LOOP
025706  010166  000012        2$:     MOV     R1,12(6)        ;SAVE THE RESULT
025712  012603                        MOV     (6)+,R3         ;POP STACK INTO R3
025714  012602                        MOV     (6)+,R2         ;POP STACK INTO R2
025716  012601                        MOV     (6)+,R1         ;POP STACK INTO R1
025720  000002                        RTI                     ;RETURN

025722                          3$:
025722  104402  025726                TYPE    .+2             ;.ASCIZ "?"<15><12>
025732  000732                        BR      4$              ;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 RUBOUT.

025734  010546                  .RDLIN: MOV     R5,-(6)         ;SAVE R5
025736  012705  026056          1$:     MOV     #INPUT,R5       ;GET ADDRESS
025742  022705  026076          2$:     CMP     #INPUT+16.,R5   ;BUFFER FULL?
025746  001423                          BEQ     4$              ;YES - TYPE "?"
025750  105737  177560                  TSTB    @#177560        ;WAIT FOR
025754  100375                          BPL     .-4             ;A CHARACTER
025756  113715  177562                  MOVB    @#177562,(5)    ;GET CHARACTER
025762  142715  000200                  BICB    #200,(5)        ;GET RID OF JUNK
025766  122715  000025                  CMPB    #25,(5)         ;IS IT A ^U
026772  001006                          BNE     5$              ;BRANCH IF NOT
026774  104402  026000                  TYPE    .+2             ;.ASCIZ "^U"<15><12>
026006  000753                          BR      1$              ;START OVER
026010  122715  000177          5$:     CMPB    #177,(5)        ;IS IT A RUBOUT
026014  001005                          BNE     3$              ;SKIP IF NOT
026016                          4$:
026016  104402  026022                  TYPE    .+2             ;.ASCIZ "?"<15><12>
026026  000743                          BR      1$              ;ZAP THE BUFFER AND LOOP
026030  111527  000000          3$:     MOVB    (5),#0          ;SET UP FOR TYPING
026034  104402  026032                  TYPE    ,3$+2           ;ECHO IT
026040  122725  000015                  CMPB    #15,(5)+        ;CHECK FOR RETURN
026044  001336                          BNE     2$              ;LOOP IF NOT RETURN
026046  104402  000012                  TYPE    ,12             ;TYPE A LINE FEED
026052  012605                          MOV     (6)+,R5         ;RESTORE R5
026054  000002                          RTI                     ;RETURN

026056  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.

026076  011646                  .TRAP:  MOV     (6),-(6)        ;GET ADDRESS OF TRAP +2
026100  162716  000002                  SUB     #2,(6)          ;MAKE IT ADDRESS OF TRAP
026104  017616  000000                  MOV     @(6),(6)        ;GET TRAP INSTRUCTION
026110  062716  121516                  ADD     #.TRP+2-TRAP,(6);GET DATA AND MAKE IT AN OFFSET
026114  013607                  .TRP:   MOV     @(6)+,PC        ;GO TO PROPER SUBROUTINE

026116  024674                          .SCOPE                  ;SCOPE  =  TRAP+0       (104400)
026120  024536                          .TYPE                   ;TYPE   =  TRAP+2       (104402)
026122  025240                          .TYPEO                  ;TYPEO  =  TRAP+4       (104404)
026124  025250                          .TYPES                  ;TYPES  =  TRAP+6       (104406)
026126  025602                          .RDOCT                  ;RDOCT  =  TRAP+10      (104410)
026130  025734                          .RDLIN                  ;RDLIN  =  TRAP+12      (104412)
026132  026176                          .CLRDK                  ;CLRDK  =  TRAP+14      (104414)
026134  026224                          .MRDMD                  ;MRDMD  =  TRAP+16      (104416)
026136  022174                          .MRCK                   ;MRCK   =  TRAP+20      (104420)
026140  022144                          .MRCLK                  ;MRCLK  =  TRAP+22      (104422)
026142  022136                          .MRINT                  ;MRINT  =  TRAP+24      (104424)
026144  022240                          .DSCK                   ;DSCK   =  TRAP+26      (104426)
026146  022222                          .MRIND                  ;MRIND  =  TRAP+30      (104430)
026150  022266                          .XBIT                   ;XBIT   =  TRAP+32      (104432)
026152  022466                          .CLKD1                  ;CLKD1  =  TRAP+34      (104434)
026154  022562                          .CLKD2                  ;CLKD2  =  TRAP+36      (104436)
026156  023444                          .CLKR1                  ;CLKR1  =  TRAP+40      (104440)
026160  023674                          .CLKR2                  ;CLKR2  =  TRAP+42      (104442)
026162  023274                          .RBIT                   ;RBIT   =  TRAP+44      (104444)
026164  026316                          .GETSP                  ;GETSP  =  TRAP+46      (104446)
026166  026356                          .SPASS                  ;SPASS  =  TRAP+50      (104450)
026170  027414                          .KBDIN                  ;KBDIN  =  TRAP+52      (104452)
026172  027274                          .SUSWR                  ;SUSWR  =  TRAP+54      (104454)
026174  027476                          .CNTLU                  ;CNTLU  =  TRAP+56      (104456)
```

```
                                        ;CLEAR ALL DISK REGISTERS
026176  012777  000040  152676  .CLRDK: MOV     #40,@RSCS2      ;CLEAR ALL DSK REG
026204  013777  001160  152670          MOV     UNNUM,@RSCS2    ;GET UNIT NUMBER
026212  005037  001174                  CLR     MCCNT           ;CLEAR MAINT CLOCK COUNT
026216  005037  001176                  CLR     MCCNT+2
026222  000002                          RTI

026224  012777  000001  152672  .MRDMD: MOV     #1,@RSMR        ;PUT DRIVE INTO MAINT MODE
026232  000002                          RTI

026234  005037  001214          WAITRY: CLR     WORK            ;CLEAR COUNTER
026240  105777  152634          1$:     TSTB    @RSCS1          ;TEST READY
026244  100406                          BMI     2$              ;OK CONT
026246  005237  001214                  INC     WORK            ;UPDATE COUNTER
026252  005737  001214                  TST     WORK            ;DONE YET?
026256  001403                          BEQ     3$              ;READY DID NOT COME UP
026260  000767                          BR      1$              ;CONTINUE WAITING
026262  062716  000002          2$:     ADD     #2,(SP)         ;UPDATE RETURN PC
026266  000207                  3$:     RTS     PC              ;RETURN

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

026270  012702  027566          MDATA:  MOV     #INBUF,R2       ;GET LEFT ADDRESS OF
026274  062702  000442                  ADD     #442,R2         ;DATA TABLE
026300  012703  000220                  MOV     #220,R3         ;SETUP COUNTER FOR 200 WORDS
026304  000241                          CLC                     ;CLEAR CARRY
026306  006142          1$:     ROL     -(R2)           ;SHIFT DATA PATTERN
026310  005303                  DEC     R3              ;DO ALL
026312  001375                  BNE     1$              ;WORDS
026314  000207                  RTS     PC

                                        ;THIS ROUTINE CLOCKS MR REG TO GET A SECTOR PULSE WHICH
                                        ;CLEARS OUT REGS. AND COUNTERS

026316  012737  002001  001202  .GETSP: MOV     #1025.,REPT     ;SETUP COUNTER
026324  104430                  MRIND           ;SEND INDEX PULSE TO MR REG
026326  104422          1$:     MRCLK           ;CLOCK MR
026330  005337  001202          DEC     REPT    ;TO REACH
026334  001374                  BNE     1$      ;SECTOR PULSE
026336  032777  000400  152560  BIT     #400,@RSMR      ;DID SECTOR PULSE SET?????
026344  001401                  BEQ     2$              ;YES
026346  000002                  RTI                     ;NO REPORT ERROR
026350  062716  000002          2$:     ADD     #2,(SP)         ;UPDATE RETURN ADDR
026354  000002                  RTI

026356  104422          .SPASS: MRCLK           ;CLOCK PAST SECTOR PULSE
026360  104422                  MRCLK
026362  005037  001174          CLR     MCCNT           ;RESET MAINT CLOCK COUNTERS
026366  005037  001176          CLR     MCCNT+2
026372  000002                  RTI
```

```
5176                                        ;ERROR TYPTXTOUT ROUTINE
5177
5178  026374  005737  025226        RSREG:  TST     .HLTCT          ;SHOULD WE TYPTXT GOOD AND BAD
5179  026400  001022                        BNE     8$              ;NO
5180  026402  104402  026406                TYPE    ,.+2            ;.ASCIZ " BAD="
5181  026414  010046                        MOV     BAD,-(6)        ;PUT BAD ON STACK
5182  026416  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5183  026420  104402  026424                TYPE    ,.+2            ;.ASCIZ " GOOD="
5184  026434  010146                        MOV     GOOD,-(6)       ;PUT GOOD ON STACK
5185  026436  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5186  026440  000402                        BR      8$              ;TYPEOUT REGISTERS
5187  026442  000137  027100                JMP     PTDONE          ;GET OUT
5188  026446                        8$:
5189  026446  104402  026452                TYPE    ,.+2            ;.ASCIZ " CS1="
5190  026460  017746  152414                MOV     @RSCS1,-(6)     ;PUT @RSCS1 ON STACK
5191  026464  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5192  026466                        1$:
5193  026466  104402  026472                TYPE    ,.+2            ;.ASCIZ " ER="
5194  026500  017746  152410                MOV     @RSER,-(6)      ;PUT @RSER ON STACK
5195  026504  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5196  026506                        2$:
5197  026506  104402  026512                TYPE    ,.+2            ;.ASCIZ " CS2="
5198  026520  017746  152356                MOV     @RSCS2,-(6)     ;PUT @RSCS2 ON STACK
5199  026524  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5200  026526  032737  000200  025226        BIT     #200,.HLTCT     ;TYPTXT SECOND SET ?
5201  026534  001076                        BNE     SEEC            ;YES
5202  026536  032737  000100  025226        BIT     #A5,.HLTCT      ;TYPTXT ER ?
5203  026544  001410                        BEQ     3$              ;NO
5204  026546  104402  026552                TYPE    ,.+2            ;.ASCIZ " A5="
5205  026560  017746  152332                MOV     @RSA5,-(6)      ;PUT @RSA5 ON STACK
5206  026564  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5207  026566  032737  000020  025226  3$:   BIT     #BA,.HLTCT      ;TYPTXT BUS ASSRESS
5208  026574  001410                        BEQ     4$              ;NO
5209  026576  104402  026602                TYPE    ,.+2            ;.ASCIZ " BA="
5210  026610  017746  152272                MOV     @RSBA,-(6)      ;PUT @RSBA ON STACK
5211  026614  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5212  026616  032737  000004  025226  4$:   BIT     #DA,.HLTCT      ;TYPTXT DA ?
5213  026624  001410                        BEQ     5$              ;NO
5214  026626  104402  026632                TYPE    ,.+2            ;.ASCIZ " DA="
5215  026640  017746  152244                MOV     @RSDA,-(6)      ;PUT @RSDA ON STACK
5216  026644  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5217  026646  032737  000010  025226  5$:   BIT     #WC,.HLTCT      ;TYPTXT WC?
5218  026654  001410                        BEQ     6$              ;NO
5219  026656  104402  026662                TYPE    ,.+2            ;.ASCIZ " WC="
5220  026670  017746  152210                MOV     @RSWC,-(6)      ;PUT @RSWC ON STACK
5221  026674  104404                        TYPEO                   ;TYPE STACK IN OCTAL
5222  026676  032737  000040  025226  6$:   BIT     #DS,.HLTCT      ;DRIVE STATUS
5223  026704  001475                        BEQ     PTDONE          ;NO
5224  026706  104402  026712                TYPE    ,.+2            ;.ASCIZ " DS="
```

```
5225  026720  017746  152166                  MOV     @RSDS,-(6)      ;PUT @RSDS ON STACK
5226  026724  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5227  026726  000137  027100                  JMP     PTDONE          ;GET OUT
5228  026732  042737  000200  025226  SEEC:   BIC     #200,.HLTCT     ;CLEAR COMMON BIT
5229  026740  032737  000240  025226          BIT     #DT,.HLTCT      ;TYPTXT DRIVE TYPE?
5230  026746  001410                          BEQ     9$              ;NO
5231  026750  104402  026754                  TYPE    .+2             ;.ASCIZ " DT="
5232  026762  017746  152140                  MOV     @RSDT,-(6)      ;PUT @RSDT ON STACK
5233  026766  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5234  026770  032737  000210  025226  9$:     BIT     #DB,.HLTCT      ;TYPTXT DATA BUFFER
5235  026776  001410                          BEQ     10$             ;NO
5236  027000  104402  027004                  TYPE    .+2             ;.ASCIZ " DB="
5237  027012  017746  152104                  MOV     @RSDB,-(6)      ;PUT @RSDB ON STACK
5238  027016  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5239  027020  032737  000220  025226  10$:    BIT     #MR,.HLTCT      ;TYPTXT MN?
5240  027026  001410                          BEQ     11$             ;NO
5241  027030  104402  027034                  TYPE    .+2             ;.ASCIZ " MR="
5242  027042  017746  152056                  MOV     @RSMR,-(6)      ;PUT @RSMR ON STACK
5243  027046  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5244  027050  032737  000204  025226  11$:    BIT     #LA,.HLTCT      ;TYPTXT LA?
5245  027056  001410                          BEQ     PTDONE          ;NO
5246  027060  104402  027064                  TYPE    .+2             ;.ASCIZ " LA="
5247  027072  017746  152022                  MOV     @RSLA,-(6)      ;PUT @RSLA ON STACK
5248  027076  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5249  027100  052737  100000  001166  PTDONE: BIS     #BIT15,ONCEE    ;SET FORND ERROR FLAG
5250  027106  032737  000040  001166          BIT     #BIT5,ONCEE
5251  027114  001466                          BEQ     1$
5252  027116  104402  027122                  TYPE    .+2             ;.ASCIZ <15><12>"MAINT CLOCK COUNT "
5253  027150  013737  001174  001226          MOV     MCCNT,WORK4     ;GET MAINT CLOCK COUNT
5254  027156  013737  001176  001222          MOV     MCCNT+2,WORK2   ;CAL NUMBERS FOR DOUBLE PRECISION
5255  027164  006137  001222                  ROL     WORK2
5256  027170  006137  001226                  ROL     WORK4
5257  027174  000241                          CLC
5258  027176  013746  001226                  MOV     WORK4,-(6)      ;PUT WORK4 ON STACK
5259  027202  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
5260  027204  012737  000005  001230          MOV     #5,WORK5
5261  027212  005037  001232          2$:     CLR     WORK6
5262  027216  006137  001222                  ROL     WORK2
5263  027222  006137  001232                  ROL     WORK6
5264  027226  006137  001222                  ROL     WORK2
5265  027232  006137  001232                  ROL     WORK6
5266  027236  006137  001222                  ROL     WORK2
5267  027242  006137  001232                  ROL     WORK6
5268  027246  013746  001232                  MOV     WORK6,-(6)      ;PUT WORK6 ON STACK
5269  027252  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
5270  027254  005337  001230                  DEC     WORK5
5271  027260  001354                          BNE     2$
5272  027262  104402  027266                  TYPE    .+2             ;.ASCIZ <15><12>
5273  027272  000207                  1$:     RTS     PC
5274
5275
5276
5277  027274  032737  000001  027412  .SUSWR: BIT     #BIT0,SWI
5278  027302  001037                          BNE     XXX
5279  027304  013746  000006                  MOV     6,-(SP)         ;SAVE 6 ON STACK
5280  027310  013746  000004                  MOV     4,-(SP)         ;SAVE 4 ON STACK
```

```
5281  027314  012737  027334  000004          MOV     #1$,4           ;SET UP TRAP ADDRESS
5282  027322  022777  177777  151476          CMP     #-1,@SWR        ;TEST 177570
5283  027330  001402                          BEQ     2$              ;FAKE OUT
5284  027332  000407                          BR      3$              ;HARDWARE AVAILABLE
5285  027334  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
5286  027336  012737  000176  001026  2$:     MOV     #SWREG,SWR      ;SET UP SOFTWARE REGISTERS
5287  027344  012737  000174  001030          MOV     #DISPREG,DISPLAY
5288  027352  022737  000176  001026  3$:     CMP     #SWREG,SWR      ;1ST TIME THRU?
5289  027360  001004                          BNE     4$              ;NO CHANGE STILL 177570
5290  027362  005737  000042                  TST     42              ;ANY XXDP OR ACT
5291  027366  001001                          BNE     4$              ;SWR=000000
5292  027370  104456                          CNTLU                   ;GET INITIAL SETTINGS
5293  027372  012637  000004          4$:     MOV     (SP)+,4         ;REPLACE 4 FROM STACK
5294  027376  012637  000006                  MOV     (SP)+,6         ;REPLACE 6 FROM STACK
5295  027402  052737  000001  027412  XXX:    BIS     #BIT0,SWI       ;SET THE BEENHEREBIT
5296  027410  000002                          RTI                     ;ALL DONE
5297
5298  027412  000000          SWI:    0
5299
5300
5301
5302  027414  005737  000042          .KBDIN: TST     42              ;GOT XXDP OR ACT
5303  027420  001057                          BNE OKT                 ;YES,GET OUT
5304  027422  022737  000176  001026          CMP     #SWREG,SWR      ;GOT SWITCH-LESS MACHINE?
5305  027430  001053                          BNE     OKT             ;NO GET OUT
5306  027432  105777  151362                  TSTB    @TKS            ;HAVE A CHARACTER
5307  027436  100050                          BPL     OKT             ;NO GET OUT
5308  027440  017737  151356  027562          MOV     @TKB,.MSG
5309  027446  042737  177600  027562          BIC     #177600,.MSG    ;STRIP OFF GARBAGE
5310  027454  122737  000007  027562          CMPB    #7,.MSG         ;DO WE HAVE A ↑G
5311  027462  001036                          BNE OKT                 ;NO,GET OUT
5312  027464  104402  027470                  TYPE    ,.+2            ;.ASCIZ <15><12>"↑G"
5313  027476                          .CNTLU:
5314  027476  104402  027502                  TYPE    ,.+2            ;.ASCIZ <15><12>"SWR= "
5315  027512  013746  000176                  MOV     SWREG,-(6)      ;PUT SWREG ON STACK
5316  027516  104404                          TYPEO                   ;TYPE STACK IN OCTAL
5317  027520  104402  027524                  TYPE    ,.+2            ;.ASCIZ "    NEW= "
5318  027536  104410                          RDOCT
5319  027540  012637  027562                  MOV     (SP)+,.MSG      ;GET NEW VALUE OFF STACK
5320  027544  005737  027564                  TST     CTN             ;DID HE TYPE <CR> OF 000000?
5321  027550  001403                          BEQ     OKT             ;DONT CHANGE IF <CR>
5322  027552  013737  027562  000176          MOV     .MSG,SWREG      ;CHANGE VALUE OF SWREG
5323  027560  000002                  OKT:    RTI                     ;ALL DONE-EXIT
5324
5325  027562  000000          .MSG:   0
5326  027564  000000          CTN:    0
5327  027566  000300          INBUF:  .BLKW   300
5328  030366  000300          OUTBUF: .BLKW   300
5329          000001                  .END
```

```
AOECK   020754          4230#
AS    = 000100           884#  1141  2212  2216  2219  2319  2323  2326  2580  3720  5202
ATA   = 100000           905#
BA    = 000020           882#  1111  1169  1175  1257  1261  1265  1287  1473  1477  5207
BAD   =%000000           828#  1011* 1234* 1273* 1274  1311* 1312  1348* 1349  1450* 1451  1487* 1488*
                        1491   1501* 1503  1517* 1518  1524* 1525  1529* 1530  1544* 1545  1552* 1553
                        1574*  1576  1598* 1599  1606* 1607  1795* 1797  1807* 1809  1873* 1875  1884*
                        1886   1896* 1898  1908* 1910  1921* 1940  1960* 1962  1971* 1973  1978* 1980
                        2017*  2019  2030* 2032  2039* 2041  2047* 2049  2131* 2132  2143* 2145  2150*
                        2153   2158* 2160  2354* 2356  2405* 2407  2459* 2461  2507* 2509  2566* 2569
                        2592*  3056* 3061  3067  3068  3070* 3073* 3075  3081  3727* 3730  3747* 3756*
                        3759   3842* 3844  4403* 4406  4415* 4418  4496* 4497  4519* 4520  4668* 4669
                        5181
BAI   = 000010           908#  4318
BEGIN   001234           791    968#
BEGIN1  000232           787    793#
BEGIN2  000226           791#   794
BELL  = 000007           802#  4344  4905
BITBA   004426          1469#
BITCS2  004312          1445#
BITST   004240          1432#
BITWC   004356          1457#
BIT0  = 000001           811#  1057  1075  3718  4653  4655  4658  4763  4766  4774  4787  5277  5295
BIT1  = 000002           812#  2003  2025  2029  2034  2036  2044  2046  2051  2054  2839  4772
BIT10 = 002000           821#  1924  1938  1955  4705  4713  4723
BIT11 = 004000           822#  4268  4662  4702
BIT12 = 010000           823#  2554  2597  2600  3063  4483  4508  4510
BIT13 = 020000           824#  1029  1040  1061  3076  4715  4720
BIT14 = 040000           780    825#  4492  4513  4515  4785
BIT15 = 100000           826#  1032  1042  1049  1502  4657  4790  4793  5249
BIT2  = 000004           813#  2081  2099  3783  3791  3795  3798  3801  3804  3807  4366  4368  4643
                        4682   4777  4780
BIT3  = 000010           814#  2087  2103  2137  2165  2167  2169  427?  4294  4638  4647
BIT4  = 000020           815#   988  1051  3304  3339  3511  3537  3882  4359
BIT5  = 000040           816#  2002  2183  2232  2244  2289  2337  2442  2617  2665  2854  2914  3054
                        3095   3145  3297  3358  3504  3555  3776  3877  3924  3971  4276  4610  4685
                        4731   4744  4752  5250
BIT6  = 000100           790    793   817#  980  2753  4059  4429  4436  4645
BIT7  = 000200           818#  2752  2764  2771  2777  4058  4070  4077  4083  4692
BIT8  = 000400           819#  2766  2776  2982  2984  3213  3215  3426  3428  3623  3625  4072  4082
                        4438   4608  4648  4663  4694
BIT9  = 001000           820#  1786  1788  1791  1929  1931  1933
CALRTB  023706          4634   4674  4679#
CHCKDV  002144          1054   1057#
CHG     022020          2148   2156  2163  4373#
CLKCNT  001206           929#  4437* 4607* 4691* 4696  4699  4703  4709  4711  4721
CLKD1 = 104434          2755   4061  5115#
CLKD2 = 104436          2759   4065  5116#
CLKR1 = 104440          2976   3207  3420  3617  5117#
CLKR2 = 104442          2978   3209  3422  3619  5118#
CLODK   023454          4636#  4676
CLRDK = 104414          1199   1432  1478  1486  1511  1536  1582  1619  1648  1825  2001  2038  2080
                        2121   2182  2231  2336  2372  2388  2423  2441  2474  2491  2521  2542  2616
                        2853   3094  3296  3503  3707  3755  3808  3826  3857  3876  3899  3921  4163
                        4242   4272  4292  4313  5107#
CNTLU = 104456          5124#  5292
```

DZRSDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
CRCTAB  022764          4528    4540#
CRCTYP  022654          3688    4528#
CRCWD   024070          4695    4720#
CS1     = 000001         878#   1106    1164    1204    1208    1212    1220    1437    1441    1496    1500    1627    1632
CS2     = 000200         885#   1097    1160    1230    1240    1244    1249    1493    1505    1522    1547    1558    1570
CTN     027564          5027*   5044*   5320    5326#
DA      = 000004         880#   1117    1332    1336    1340    1361    2371    2422    2575    2591    2832    2838    3042
                        3048    3274    3280    3483    3680    3792    3799    3805    3892    3896    3902    3906    4142
                        4151    5212
DAO     = 001000         906#
DB      = 000210         887#   5234
DCK     = 100000         907#
DISPLA  001030           843#   4883*   4887*   5287*
DISPRE  000174           783#   5287
DLT     = 100000         900#   1568
DONE    021570          1070    4338#
DONEE   002320          1052    1058    1070#
DRY     = 000200         901#
DS      = 000040         883#   1101    2027    2094    2098    2109    2113    2212    2216    2219    2319    2323    2326
                        2366    2375    2379    2417    2426    2430    2471    2477    2481    2518    2524    2528    2584
                        3042    3274    3483    3680    3686    3720    3738    3792    3799    3805    3811    3851    3855
                        3861    3865    3887    3892    3902    3906    4134    4148    4237    4240    4245    4248    4290
                        4305    4327    4331    5222
DSCK    = 104426        2194    2200    2207    2269    2275    2314    2344    2361    2395    2412    2449    2466    2498
                        2513    4200    5112#
DT      = 000240         889#   5229
DVNUM   001724          1023    1026#
ER      = 000002         879#   1123    1181    1369    1373    1380    1386    1393    3738
ERR     = 040000         904#
ERRORS  001002           833#   4906*   4926
E1910   023256          4568    4595#
E1915   023265          4570    4597#
E192    023242          4564    4593#
E197    023250          4566    4594#
E2410   023215          4558    4587#
E2412   023224          4560    4589#
E2415   023233          4562    4591#
E242    023173          4552    4581#
E245    023201          4554    4583#
E247    023207          4556    4585#
E3010   023146          4546    4575#
E3012   023155          4548    4577#
E3015   023164          4550    4579#
E302    023124          4540    4572#
E305    023132          4542    4573#
E307    023140          4544    4574#
FILCHR  001014           837#   4831    4832
FLAG2   001144           790*    793*    912#    978*    980    2615*   2839    2862*   3103*   3295*   3304    3339*   3502*
                        3511    3537*   3920*   4271*   4610    4643    4653    4655*   4658*   4731    4744    4752
FLOTBA  003350          1270#
FLOTDA  003650          1345#
FLOTWC  003510          1308#
GENCRC  024112          2778    2986    3217    4089    4730#
GETSP   = 104446        2651    2899    3131    3343    3541    3957    5120#
GOOD    =%000001         827#   1012*   1235*   1270#   1272    1274    1277*   1308*   1310    1312    1315*   1345*   1347
                        1349    1352*   1448*   1449*   1451    1489*   1490*   1491    1502*   1503    1515*   1516*   1518
```

DZRSDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    1523*  1525   1528*  1530   1542*  1543*  1545   1548*  1549*  1550   1553   1575*  1576
                    1584*  1585   1586*  1589*  1591   1592*  1597*  1599   1602*  1605*  1607   1611*  1796*
                    1797   1808*  1809   1874*  1875   1885*  1886   1897*  1898   1909*  1910   1939*  1940
                    1961*  1962   1972*  1973   1979*  1980   2018*  2019   2031*  2032   2040*  2041   2048*
                    2049   2129*  2130*  2132   2144*  2145   2151*  2152*  2153   2159*  2160   2355*  2356
                    2406*  2407   2460*  2461   2508*  2509   2551*  2552   2556*  2561   2562   2567*  2569
                    2593*  3055*  3061   3067   3069   3071*  3074*  3081   3728*  3729*  3730   3748*  3757*
                    3758*  3759   3843*  3844   4404*  4406   4416*  4418   4493*  4494*  4497   4516*  4517*
                    4520   4642*  4645*  4646*  4647*  4652*  4657*  4662*  4667*  4669   4702*  4705*  4713*
                    4715*  4720*  4723*  5184
HLT    = 104000      799#  1013   1097   1101   1106   1111   1117   1123   1129   1135   1141   1160   1164
                    1169   1175   1181   1186   1191   1204   1208   1212   1220   1230   1236   1240   1244
                    1249   1257   1261   1265   1276   1287   1295   1299   1303   1314   1324   1332   1336
                    1340   1351   1361   1369   1373   1380   1386   1393   1403   1410   1418   1426   1437
                    1441   1453   1461   1465   1473   1477   1493   1496   1500   1505   1522   1527   1532
                    1547   1555   1558   1570   1578   1601   1609   1627   1632   1659   1665   1677   1681
                    1690   1694   1697   1706   1710   1719   1723   1733   1737   1743   1750   1754   1765
                    1779   1783   1799   1811   1838   1846   1849   1853   1856   1865   1869   1877   1888
                    1900   1912   1942   1964   1975   1982   1985   2026   2027   2035   2045   2053   2094
                    2098   2109   2113   2135   2149   2157   2164   2196   2202   2209   2212   2216   2219
                    2241   2248   2252   2259   2263   2271   2277   2284   2293   2297   2306   2310   2316
                    2319   2323   2326   2346   2360   2363   2366   2371   2375   2379   2397   2411   2414
                    2417   2422   2426   2430   2451   2465   2468   2471   2477   2481   2500   2512   2515
                    2518   2524   2528   2571   2575   2580   2584   2591   2594   2654   2660   2669   2673
                    2682   2686   2694   2698   2706   2710   2714   2720   2724   2733   2737   2744   2758
                    2763   2794   2802   2806   2815   2819   2826   2832   2835   2838   2902   2909   2918
                    2922   2931   2935   2942   2946   2952   2960   2965   2977   2979   2999   3003   3007
                    3011   3021   3025   3035   3042   3045   3048   3072   3134   3140   3149   3153   3162
                    3166   3174   3178   3184   3191   3196   3208   3210   3230   3234   3238   3242   3252
                    3256   3266   3274   3277   3280   3346   3353   3362   3366   3375   3379   3386   3390
                    3396   3404   3409   3421   3423   3439   3443   3447   3451   3461   3465   3475   3483
                    3486   3544   3550   3559   3563   3572   3576   3583   3587   3593   3601   3606   3618
                    3620   3636   3640   3644   3648   3658   3662   3672   3680   3683   3686   3720   3732
                    3738   3742   3749   3761   3792   3799   3805   3811   3847   3851   3855   3861   3865
                    3887   3892   3896   3902   3906   3960   3966   3975   3979   3988   3992   4000   4004
                    4012   4016   4020   4026   4030   4039   4043   4050   4064   4069   4103   4110   4114
                    4122   4126   4132   4134   4142   4145   4148   4151   4202   4211   4218   4222   4234
                    4237   4240   4245   4248   4287   4290   4302   4305   4327   4331
HLTADR  001012       836#  4910*  4911*  4912   4913   4927
IADONE  017102      3786   3813#
ICNT    001000       832#   975*  4358*  4861   4876   4878   4880*  4881*  4883   4886*  4887   4893   4922*
IE     = 000100       909#
ILFDON  007326      2071   2073#
INBIT   001210       930#  4756*  4758*  4760*  4764   4767   4775*  4778   4788*  4791   4807   4808*  4809*
INBUF   027566      2631   2648   2745   2780   2870   2969   2988   3055   3111   3128   3200   3219   3321
                    3413   3429   3520   3610   3626   3936   3954   4051   4091   4278*  4280   4295   4319
                    4320*  4734   5148   5327#
INFTST  021564      4270   4335#
INPUT   026056       986   5028   5063   5064   5087#
INT     005176      1619#
INTDON  005266      1628   1633#
INTMR   020460      3952   4140#
INTMR1  020530      4135   4152#
IR     = 000100       895#
KBDIN  = 104452      4858   4902   5122#
LA     = 000204       886#  1665   1697   1765   1849   1856   5244
```

DZRSDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
LAD     001010          835#   976*  3778*  4357*  4882*  4888  4890  4893
LBT   = 002000          903#
LSTEV   001146          913#  4431*  4445*  4490
LSTOD   001150          914#  4432*  4444*  4479
LSTWD   024034         4693   4709#
MCCNT   001174          925#  1828*  4208*  4399*  4400*  4640*  4641*  5128*  5129*  5173*  5174*  5253  5254
MDATA   026270         3489   3692   5148#
MPRO  = 172100          923#
MR    = 000220          888#  1129   1186   1403   1410   1418   1426   1665   1697   1765   2654   2902  3134
                       3346   3544   3960   5239
MRAOE   020534         4163#
MRCAL   023744         4495   4518   4691#
MRCILF  007024         2038#
MRCK  = 104420         1651   1657   1675   1679   1688   1692   1704   1708   1717   1721   1731   1735  1741
                       1748   1752   1777   1781   1831   1836   1844   1851   1863   1867   1983   2009  2083
                       2123   2185   2234   2239   2246   2250   2257   2261   2282   2291   2295   2304  2308
                       2339   2390   2444   2493   2544   2620   2658   2667   2671   2680   2684   2692  2696
                       2704   2708   2712   2718   2722   2731   2735   2742   2792   2800   2804   2813  2817
                       2824   2857   2907   2916   2920   2929   2933   2940   2944   2950   2958   2963  2997
                       3001   3005   3009   3019   3023   3033   3098   3138   3147   3151   3160   3164  3172
                       3176   3182   3189   3194   3228   3232   3236   3240   3250   3254   3264   3300  3351
                       3360   3364   3373   3377   3384   3388   3394   3402   3407   3437   3441   3445  3449
                       3459   3463   3473   3507   3548   3557   3561   3570   3574   3581   3585   3591  3599
                       3604   3634   3638   3642   3646   3656   3660   3670   3709   3780   3831   3879  3927
                       3964   3973   3977   3986   3990   3998   4002   4010   4014   4018   4024   4028  4037
                       4041   4048   4101   4108   4112   4120   4124   4130   4166   4209   4216   4220  4232
                       5109#
MRCLK = 104422         1674   1678   1687   1691   1703   1707   1716   1720   1730   1734   1740   1747  1751
                       1773   1776   1780   1804   1843   1850   1862   1866   1883   1894   1907   1917  1959
                       1970   2199   2206   2245   2249   2256   2260   2265   2274   2290   2294   2303  2307
                       2313   2666   2670   2679   2683   2691   2695   2703   2707   2711   2717   2721  2730
                       2734   2741   2790   2799   2803   2812   2816   2823   2915   2919   2928   2932  2939
                       2943   2949   2996   3000   3004   3008   3018   3022   3032   3039   3146   3150  3159
                       3163   3171   3175   3181   3227   3231   3235   3239   3249   3253   3263   3271  3359
                       3363   3372   3376   3383   3387   3393   3436   3440   3444   3448   3458   3462  3472
                       3480   3556   3560   3569   3573   3580   3584   3590   3633   3637   3641   3645  3655
                       3659   3669   3677   3972   3976   3985   3989   3997   4001   4009   4013   4017  4023
                       4027   4036   4040   4047   4100   4107   4111   4119   4123   4129   4199   4215  4219
                       4230   4231   4393   4394   4474   4505   5110#  5162   5171   5172
MRCRC   014664         3296#  3490
MRCRC1  015070         3359#  3368
MRCRC2  015144         3383#  3392
MRCRC3  015246         3412#
MRCRC4  015426         3456#
MRCRC5  015472         3480#
MRDCK   015544         3503#  3693
MRDCK1  015750         3556#  3565
MRDCK2  016024         3580#  3589
MRDCK3  016126         3609#
MRDCK4  016306         3653#
MRDCK5  016352         3677#
MRDMD = 104416         2008   2082   2122   2184   2233   2338   2389   2443   2492   2543   3779   5108#
MRDSEL  011540         2542#
MRD3    013346         2968#
MRD4    013546         3017#
MRD5    013612         3039#
```

```
MRD6    013650          3054#
MREX    017546          3921#
MREX1   017742          3972#   3981
MREX2   020016          3997#   4006
MREX3   020102          4023#   4032
MRIFT   016466          3707#
MRILF   006606          2001#
MRIND = 104430          1650    1655    1830    1834    1976    2015    2126    2188    2238    2231    2619    2657    2856
                        2906    3097    3137    3299    3350    3506    3547    3708    3830    3837    3838    3839    3878
                        3926    3963    4165    4182    4207    5113#   5161
MRINT = 104424          1653    1833    2011    2085    2125    2187    2236    2341    2392    2446    2495    2546    2622
                        2859    3100    3302    3509    3711    3782    3833    3881    3929    4168    5111#
MRLF1   006632          2004    2008#   2072
MRLF2   006644          2016#
MRLF3   007244          2057#   2059    2061    2063    2065    2067    2069
MROP    007330          2080#
MROPER  007520          2121#   2168
MROPI   017104          3826#
MROPIA  017276          3854    3857#
MRPAR   017420          3876#
MRRD    013000          2853#
MRRD1   013170          2915#   2924
MRRD2   013244          2939#   2948
MRRT1   010320          2245#   2254
MRSAC   010254          2231#
MRSRCH  010072          2182#
MRTIME  005270          1648#
MRTIM1  005336          1674#   1683
MRT2    005414          1696    1701#
MRT2A   005424          1703#   1712
MRT2B   005460          1716#   1725
MRT2C   005514          1730#   1739
MRT3    005632          1772#   1802
MRT4    006024          1825#
MRT4A   006076          1843#   1858
MRT4B   006154          1866#   1952
MRT4C   006206          1883#   1890
MRT4D   006260          1906#   1944
MRT4E   006266          1907#   1914    1956
MRT4F   006506          1919    1958#
MRT4G   006540          1970#
MRVR    021040          4268#
MRVRR   021350          4304    4310#
MRVR1   021200          4289    4292#
MRVR2   021354          2841    4313#
MRWCK   014020          3094#
MRWCK1  014174          3146#   3155
MRWCK2  014250          3171#   3180
MRWCK3  014352          3199#
MRWCK4  014552          3248#
MRWCK5  014616          3271#
MRWRT   012060          2616#   4310
MRWRT1  012226          2666#   2675
MRWRT2  012302          2691#   2700
MRWRT3  012366          2717#   2726
MRWR1   010470          2290#   2299
```

```
MULSYS  021652          4349   4355#  5008
MULTII  001674           982    987  1019#  4361
N     = 000070           798#  1078   1082#  1142   1146#  1192   1196#  1213   1214#  1222   1226#  1251   1252#
                        1267   1268#  1279   1280#  1288   1289#  1304   1305#  1318   1319#  1325   1326#  1341
                        1342#  1355   1356#  1362   1363#  1375   1376#  1381   1382#  1387   1388#  1394   1395#
                        1404   1405#  1411   1412#  1419   1420#  1427   1431#  1443   1444#  1455   1456#  1467
                        1468#  1479   1483#  1506   1510#  1533   1534#  1579   1580#  1615   1619#  1634   1638#
                        1812   1816#  1986   1990#  2074   2078#  2115   2119#  2170   2174#  2220   2224#  2327
                        2331#  2380   2384#  2432   2436#  2482   2486#  2529   2533#  2602   2606#  2842   2846#
                        3084   3088#  3281   3285#  3492   3496#  3695   3699#  3764   3768#  3814   3818#  3866
                        3870#  3908   3912#  4153   4157#  4261   4265#
NEC   = 010000           898#  2582
NEDOON  011760          2558   2597#
NMOD    012050          2590   2595   2598   2601#
NOPERR  021724          2093   2097   2108   2112   4366#
NOWEV   001152           915#  4441*  4445   4454*  4456*  4464*  4466*  4487   4511   4615*  4621*  4623*  4630*
                        4632*  4683
NOWGO   002324          1016   1050   1069   1075#  4362
NOWOD   001154           916#  4440*  4444   4451*  4453*  4461*  4463*  4476   4506   4614*  4618*  4620*  4627*
                        4629*  4680
OKT     027560          5303   5305   5307   5311   5321   5323#
ONCEE   001166           921#   979*   988*  1032*  1042   1049*  1051   1057   1075*  1649*  1786   1788*  1791*
                        1826*  1827*  1924   1929   1931*  1933*  1938*  1955*  2002*  2003   2025*  2029*  2034*
                        2036*  2044*  2046*  2051*  2054*  2081*  2087   2099*  2103*  2114*  2137   2165   2167*
                        2169*  2183*  2232*  2244*  2289*  2337*  2442*  2597   2600*  2617*  2618*  2665*  2752*
                        2753*  2764   2766   2771*  2776*  2777*  2854*  2855*  2914*  2982   2984*  3054*  3095*
                        3096*  3145*  3213   3215*  3297*  3298*  3358*  3426   3428*  3504*  3505*  3555*  3623
                        3625*  3776*  3783   3791*  3795*  3798*  3801*  3804*  3807*  3877*  3924*  3925*  3971*
                        4058*  4059*  4070   4072   4077*  4082*  4083*  4276*  4359   4366   4368*  4429   4436*
                        4438   4608   4648   4663   4692   4694   5249*  5250
OR    = 000200           896#  1556
OUTBUF  030366          2891   2895   3056   3334   3338   3532   3536   3715   3745   3748   3827   4172   4175
                        5328#
PC    =%000007           810#  2093*  2097*  2108*  2112*  2148*  2156*  2163*  2778*  2986*  3217*  3489*  3688*
                        3692*  4089*  4372*  4376*  4495*  4518*  4535*  4537*  4541*  4543*  4545*  4547*  4549*
                        4551*  4553*  4555*  4557*  4559*  4561*  4563*  4565*  4567*  4569*  4571*  4634*  4674*
                        4686*  4706*  4716*  4724*  4762*  4776*  4789*  4802*  4810*  4916*  5099*  5143*  5155*
                        5273*
PCNT    001004           834#  4339*  4340*  4351
PGE   = 002000           897#
PGTRAP  005252          1620   1629#
PIP   = 020000           902#
PS    = 177776           800#   801   1082*  1147*  1622*  3922*
PSW   = 177776           801#
PTDONE  027100          5187   5223   5227   5245   5249#
QQ    = 000001          1078#  1142#  1192#  1213#  1222#  1251#  1267#  1279#  1288#  1304#  1318#  1325#  1341#
                        1355#  1362#  1375#  1381#  1387#  1394#  1404#  1411#  1419#  1427#  1443#  1455#  1467#
                        1479#  1506#  1533#  1579#  1615#  1634#  1812#  1986#  2074#  2115#  2170#  2220#  2327#
                        2380#  2432#  2482#  2529#  2602#  2842#  3084#  3281#  3492#  3695#  3764#  3814#  3866#
                        3908#  4153#  4261#
RBIT  = 104444          2975   3206   3419   3616   5119#
RDLIN = 104412           985   5025   5106#
RDOCT = 104410           991   5105#  5318
REGCHG  022034          4373   4378#
REPT    001202           927#  1673*  1682*  1702*  1711*  1715*  1724*  1729*  1738*  1772*  1774*  1803*  1805*
                        1842*  1857*  1882*  1889*  1905*  1943*  1954*  1958*  1965*  2198*  2203*  2243*  2253*
```

```
                    2264*  2266*  2273*  2278*  2288*  2298*  2664*  2674*  2690*  2699*  2716*  2725*  2798*
                    2807*  2913*  2923*  2938*  2947*  2972*  2980*  2994*  3017*  3026*  3057*  3065*  3144*
                    3154*  3170*  3179*  3203*  3211*  3225*  3248*  3257*  3357*  3367*  3382*  3391*  3416*
                    3424*  3434*  3456*  3466*  3554*  3564*  3579*  3588*  3613*  3621*  3631*  3653*  3663*
                    3970*  3980*  3996*  4005*  4022*  4031*  4106*  4115*  4176*  4178*  4186*  4194*  4214*
                    4223*  4281*  4285*  4296*  4300*  4650   4730*  4733*  4796*  5160*  5163*
REPT1   001204       928#  1771*  1801*  1906*  1913*  1953*  2971*  2985   2989*  3202*  3216   3220*  3415*
                    3612*  4659*  4661*  4665   4743*  4746*  4754   4794*
RMRC1   010612      2336#
RMRC2   010776      2388#
RMRC3   011152      2441#
RMRC4   011324      2491#
RSAS    001116       856#  1009   1011   1026   1139   2131   2150   2213*  2214   2320*  2321   2504*  2578
                    3713*  5205
RSBA    001106       852#  1085*  1109   1150*  1167   1254*  1255   1258*  1259   1262*  1263   1272*  1273
                    1283*  1284*  1285   1469*  1471   1474*  1475   2648*  2891*  3128*  3334*  3532*  3715*
                    3745   3747   3827*  3954*  4172*  4280*  4295*  4319*  5210
RSBAB   001142       866#  1470*
RSCS1   001100       849#  1084*  1104   1149*  1162   1201*  1202   1205*  1206   1209*  1210   1216*  1217*
                    1218   1433*  1435   1438*  1439   1494   1497*  1498   1623*  1630   2016*  2089*  2105*
                    2139*  2142*  2193*  2210   2217   2268*  2317   2324   2343*  2350*  2364   2377   2394*
                    2415   2428   2448*  2469   2479   2497*  2516   2526   2560*  2573   2650*  2830   2893*
                    3040   3130*  3272   3336*  3481   3534*  3678   3717*  3718   3736   3788*  3802   3835*
                    3853   3890   3900   3956*  4140   4181*  4282*  4283   4288   4297*  4298   4303   4321*
                    4322   4329   5136   5190
RSCS1B  001134       863#  1434*  2581*
RSCS2   001102       850#  1005*  1020*  1024*  1035*  1083*  1090*  1095   1098*  1148*  1155*  1157*  1158
                    1161*  1228   1231*  1232   1234   1237*  1238   1241*  1242   1245*  1246*  1247   1250*
                    1445*  1446*  1450   1488   1501   1517   1544   1556   1568   2552*  2554   2562*  2566
                    2582   2586*  3727   3756   3882*  4279*  4294*  4318*  4324   5126*  5127*  5198
RSCS2B  001136       864#  1447*
RSDA    001110       853#  1086*  1115   1151*  1173   1329*  1330   1333*  1334   1337*  1338   1347*  1348
                    1357*  1358*  1359   2189*  2192*  2342*  2369   2393*  2401*  2420   2447*  2496*  2548*
                    2563*  2587   2836   3046   3278   3787*  3883*  3894   4149   4170*  5215
RSDB    001122       858#  1153*  1487   1512*  1513*  1524   1529   1539*  1552   1564*  1574   1585*  1591*
                    1598   1606   5237
RSES    001112       854#  1099   2017   2039   2095   2110   2158   3796   3849   3862   4238   4246   4415
                    5225
RSDT    001126       860#  1036   1038   1758   1784   1922   2190   5232
RSER    001114       855#  1006*  1021*  1087*  1121   1152*  1179   1366*  1367   1370*  1371   1377*  1378
                    1383*  1384   1389*  1390*  1391   2030   2047   2091   2106   2128*  2143   2354   2373
                    2405   2424   2455*  2459   2475   2507   2522   2553   3684   3712*  3789   3809   3842
                    3858   3885   3904   4088*  4146   4235   4243   5194
RSLA    001120       857#  1663   1695   1763   1795   1807   1847   1854   1873   1884   1896   1908   1921
                    1960   1971   1978   5247
RSMR    001124       859#  1088*  1127   1156*  1184   1398*  1399   1406*  1407*  1408   1413*  1414   1421*
                    1422   2956*  2957*  2961*  2962*  3187*  3188*  3192*  3193*  3400*  3401*  3405*  3406*
                    3597*  3598*  3602*  3603*  4190*  4191*  4397*  4398*  4403   4412*  4413*  4496   4519
                    4637*  4639*  4668   5132*  5165   5242
                    4916   5178#
RSREG   026374
RSVCPS  001132       862#  1621*  3953*
RSVEC   001130       861#  1620*  3952*
RSWC    001104       851#  1089*  1133   1154*  1189   1292*  1293   1296*  1297   1300*  1301   1310*  1311
                    1320*  1321*  1322   1457*  1459   1462*  1463   2649*  2833   2892*  3043   3129*  3275
                    3335*  3484   3533*  3681   3716*  3740   3828*  3955*  4143   4171*  4277*  4293*  4317*
                    5220
```

```
RSWCB   001140          865# 1458*
RS0     001156          917# 4767* 4775  4788
R0    =%000000          803# 4344* 4985  5004*
R1    =%000001          804# 1019* 1022* 1537* 1538* 1539  1540  1562* 1563* 1564  1565  4986  4996*
                       4997* 5003* 5022  5026* 5039* 5041* 5043* 5046  5049*
R2    =%000002          805#  996*  998* 1003  1004  1701* 1760* 1762* 1763  1789* 1792* 1794* 1796
                       1829* 1874  1885  1895* 1897  1909  1918  1920* 1934* 1936* 1937* 1939  1948
                       1961  2005* 2016  2022  2057* 2058  2060  2062  2064  2066  2068  2070  2631*
                       2632* 2633* 2638* 2643* 2780* 2781* 2782* 2783  2870* 2871* 2872* 2877* 2881*
                       2894* 2897* 2988* 2990* 2991* 2992  3111* 3112* 3113* 3118* 3122* 3219* 3221*
                       3222* 3223  3321* 3323* 3326* 3327* 3329* 3337* 3341* 3429* 3430* 3431* 3432
                       3520* 3522* 3525* 3526* 3528* 3535* 3539* 3626* 3627* 3628* 3629  3777* 3785*
                       3787  3936* 3937* 3938* 3943* 3947* 4091* 4092* 4093* 4094  4188* 4190  4987
                       5002* 5023  5029* 5030  5032  5034  5036* 5037* 5038* 5040* 5042* 5048* 5148*
                       5149* 5152*
R3    =%000003          806# 1948* 1949* 1950  2634* 2636* 2638  2640* 2644* 2873* 2875* 2877  2879*
                       2882* 2895* 2896* 3068* 3070  3073  3114* 3116* 3118  3120* 3123* 3322* 3324*
                       3328* 3330* 3338* 3340* 3521* 3523* 3527* 3529* 3536* 3538* 3939* 3941* 3943
                       3945* 3948* 4189* 4191  4475* 4483* 4492* 4494  4508* 4510* 4513* 4515* 4517
                       4635* 4636* 4637  4638* 4639  4646  4675* 4806* 4807* 4809  4988  5001* 5024
                       5028* 5047* 5150* 5153*
R4    =%000004          807#  992*  993  995   999  1001* 2642* 2643  2746* 2880* 2881  3069* 3071
                       3074  3121* 3122  3946* 3947  4052* 4187* 4192* 4449* 4452* 4455* 4462* 4465*
                       4606* 4619* 4622* 4628* 4631* 4735* 4757* 4799* 4821  4831* 4835* 4849* 4940
                       4943* 4963  4969* 4989  5000*
R5    =%000005          808# 2745* 2746  2783* 2784* 2969* 2970* 2992* 2993* 3058* 3059* 3200* 3201*
                       3223* 3224* 3413* 3414* 3432* 3433* 3610* 3611* 3629* 3630* 4051* 4052  4094*
                       4095* 4175* 4177* 4448* 4449  4605* 4606  4734* 4735  4798* 4799  4822  4823*
                       4824  4826* 4827  4829  4837  4845* 4846* 4847  4848* 4941  4942* 4949* 4951*
                       4953* 4968* 4990  4999* 5062  5063* 5064  5084*
SAVEE   001172          924# 2985* 2989  3216* 3220  3306* 3431  3487* 3513* 3690* 4530
SC    = 100000          894#
SCOPE = 104400         1081  1145  1195  1213  1225  1251  1267  1279  1288  1304  1318  1325  1341
                       1355  1362  1375  1381  1387  1394  1404  1411  1419  1430  1443  1455  1467
                       1482  1509  1533  1579  1618  1637  1815  1989  2077  2118  2173  2223  2330
                       2383  2435  2485  2532  2605  2845  3087  3284  3495  3698  3767  3817  3869
                       3911  4156  4264  4338  5101#
SEEC    026732         5201  5228#
SILO    004710         1536#
SILOB   004500         1486#
SILOFL  005062         1582#
SP    =%000006          809#  968* 3923* 4164* 4404  4405* 4408* 4416  4417* 4420* 4499* 4522* 4671*
                       4991  4995* 5142* 5168* 5279* 5280* 5285  5293  5294  5319
SPASS = 104450         2655  2903  3135  3347  3545  3961  5121#
STTEST  002010         1030  1033# 1056
SUSWR = 104454          977  5123#
SWI     027412         5277  5295* 5298#
SWR     001026          842# 1029  1040  1061  3063  3076  4268  4341  4859  4861  4863  4874  4903
                       4907  4917  4920  5282  5286* 5288  5304
SWREG   000176          784# 5286  5288  5304  5315  5322*
SW10  = 002000          772# 4341  4902  4903
SW11  = 004000          771# 4874
SW12  = 010000          770# 4342  4344
SW13  = 020000          769# 4907
SW14  = 040000          768# 4863
SW15  = 100000          767#
```

```
SWB    = 000400            775#   4858    4859
SW9    = 001000            774#   4920
TBDIA    021266           4291    4308#
TIMES    025066           1076    1077*   1200*   4878    4893#
TIMSV    001170            922#   1076*   1200
TKB      001022            840#   5308
TKS      001020            839#   5306
TPB      001024            841#   4832*   4837*
TPS      001016            838#   4833    4838
TRE    = 040000            893#   1497
TRMR     022071           2358    2409    2463    4384#
TRY      001704           1021#   1025
TRYNX    002120           1034    1039    1051#   4335
TSTEVB   022512           4477    4480    4487#
TST1     002346           1081#
TST10    003400           1279#
TST11    003424           1288#
TST12    003506           1304#
TST13    003540           1318#
TST.4    003564           1325#
TST15    003646           1341#
TST16    003700           1355#
TST17    003724           1362#
TST2     002570           1145#
TST20    003766           1375#
TST21    004010           1381#
TST22    004032           1387#
TST23    004056           1394#
TST24    004114           1404#
TST25    004142           1411#
TST26    004200           1419#
TST27    004236           1430#
TST3     003004           1195#
TST30    004310           1443#
TST31    004354           1455#
TST32    004424           1467#
TST33    004476           1482#
TST34    004600           1509#
TST35    004706           1533#
TST36    005060           1579#
TST37    005174           1618#
TST4     003076           1213#
TST40    005266           1637#
TST41    006022           1815#
TST42    006604           1989#
TST43    007326           2077#
TST44    007516           2118#
TST45    010070           2173#
TST46    010252           2223#
TST47    010610           2330#
TST5     003124           1225#
TST50    010774           2383#
TST51    011150           2435#
TST52    011322           2485#
TST53    011536           2532#
TST54    012050           2605#
```

DZRSDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TST55   012776          2845#
TST56   014016          3087#
TST57   014654          3284#
TST6    003264          1251#
TST60   015534          3495#
TST61   016464          3698#
TST62   016704          3767#
TST63   017102          3817#
TST64   017416          3869#
TST65   017536          3911#
TST66   020532          4156#
TST67   021036          4264#
TST7    003346          1267#
TTAGG   002572          1147#
TYPE  = 104402           984    990   1007   1031   1044   1048   1065   1066   2021   2043   2052   2134   2147
                        2155   2162   2358   2359   2409   2410   2463   2464   2511   2599   3078   3687   3846
                        3860   4309   4326   4343   4356   4369   4373   4536   4540   4542   4544   4546   4548
                        4550   4552   4554   4556   4558   4560   4562   4564   4566   4568   4570   4905   4909
                        4915   4967   5007   5053   5072   5077   5080   5083   5102#  5180   5183   5189   5193
                        5197   5204   5209   5214   5219   5224   5231   5236   5241   5246   5252   5272   5312
                        5314   5317
TYPE0 = 104404          4914   5103#  5182   5185   5191   5195   5199   5206   5211   5216   5221   5226   5233
                        5238   5243   5248   5316
TYPES = 104406          1047   1064   2024   3080   4371   4375   5104#  5259   5269
UNCMP   001164           920#  1004*  1027*  1033   1053*  1059*  1067*  2129   2151
UNITSV  001162           919#  1003*  1009   1012   1026*  1033   2213   2320   3713
UNNUM   001160           918#   995*  1005   1028*  1035   1046   1055*  1060*  1063   1083   1098   1148   1161
                        1250   1445   1448   1490   1516   1543   2586   2593   3729   3758   5127
WAITRY  026234          5135#
WC    = 000010           881#  1135   1191   1295   1299   1303   1324   1461   1465   2832   2835   3042   3045
                        3274   3277   3483   3486   3680   3683   3686   3742   3855   4134   4142   4145   4148
                        4290   4305   5217
WCE   = 040000           899#  4324
WCRC    001200           926#  4747*  4772   4785
WK15    001212           931#  4748*  4751*  4761
WORK    001214           932#  1399*  1400*  1401   1414*  1415*  1416   1422*  1423*  1424   1514*  1520*  1624*
                       1625*  2022*  2023   2090*  2104*  2136*  2141*  2550*  2557*  2747*  2768*  2772*  2773*
                        2782   2785*  2786*  2991   3075*  3079   3222   3722*  3723*  4053*  4074*  4078*  4079*
                        4093   4096*  4097*  4314*  4315*  4370   4374   4528*  4532*  4535   4636   4679*  4682*
                       4685*  4761*  4771*  4774*  4784*  4787*  4801*  4806   4808   5135*  5138*  5139
WORK0   001216           933#  4736*  4747   4750*  4763*  4766*  4777*  4780*  4790*  4793*  4801
WORK1   001220           934#  2561*  2592   4529*  4530   4533*
WORK2   001222           935#  5254*  5255*  5262*  5264*  5266*
WORK3   001224           936#  2968*  3199*  3412*  3609*  4442*  4446   4457*  4603   4612*  4616*  4624*
WORK4   001226           937#  5253*  5256*  5258
WORK5   001230           938#  4273*  4274*  5260*  5270*
WORK6   001232           939#  5261*  5263*  5265*  5267*  5268
XBIT  = 104432          2754   4060   5114#
XXOR    024512          4762   4776   4789   4806#
XXX     027402          5278   5295#
ZERONE  004602          1511#
$ENDAD  021634           778   4347#
$END1   021644          4345   4349#
.     = 031166           776#   777#   779#   782#   786#   789#   830#   845#   984    985#   990    1007   1008#
                        1031   1044   1066   1067#  1096   1100   1105   1110   1116   1122   1128   1134   1140
                        1159   1163   1168   1174   1180   1185   1190   1203   1207   1211   1219   1229   1239
```

```
                    1243    1248    1256    1260    1264    1275    1286    1294    1298    1302    1313    1323    1331
                    1335    1339    1350    1360    1368    1372    1379    1385    1392    1402    1409    1417    1425
                    1436    1440    1452    1460    1464    1472    1476    1492    1495    1499    1504    1526    1531
                    1546    1557    1566    1569    1577    1587    1600    1603    1608    1631    1664    1810    1848
                    1855    2021    2043    2044*   2052    2053*   2134    2135*   2147    2155    2162    2218    2325
                    2359    2378    2410    2411*   2429    2464    2465*   2480    2511    2512*   2527    2570    2574
                    2579    2583    2599    2834    2837    2840    3044    3047    3078    3276    3279    3485    3682
                    3687    3688*   3741    3760    3810    3846    3860    3864    3886    3891    3895    3901    3905
                    4141    4144    4147    4150    4247    4309    4326    4327*   4343    4344*   4351    4356    4369
                    4391*   4536    4834    4839    4893    4909    4910*   4915    4916*   4918    4926    4927    4936
                    4956    4959    4964    4974*   5007    5011    5053    5067    5072    5073*   5077    5087*   5180
                    5183    5184*   5189    5193    5194*   5197    5204    5205*   5209    5210*   5214    5215*   5219
                    5220*   5224    5225*   5231    5232*   5236    5237*   5241    5242*   5246    5247*   5252    5253*
                    5272    5273*   5312    5313*   5314    5317    5318*   5327*   5328*
.CLKD1  022466      4474*   5115
.CLKD2  022562      4505*   5116
.CLKR1  023444      4634*   5117
.CLKR2  023674      4674*   5118
.CLRDK  026176      5107    5126*
.CNTLU  027476      5124    5313*
.DSCK   022240      4415*   5112
.GETSP  026316      5120    5160*
.HLT    025070       971    4902*
.HLTCT  025226      4912*   4926*   5178    5200    5202    5207    5212    5217    5222    5228*   5229    5234    5239
                    5244
.ILLUP  025570      4983    5010*
.KBDIN  027414      5122    5302*
.KIT    025040      4864    4873    4879    4886*   4924
.MRCK   022174      4403*   5109
.MRCLK  022144      4397*   5110
.MRDMD  026224      5108    5132*
.MRIND  022222      4412*   5113
.MRINT  022136      4393*   5111
.MSG    027562      5308*   5309*   5310    5319*   5322    5325*
.OVER   025044      4862    4887*
.POWER  025442       969    4983*   5005
.POWUP  025506      4992    4995*
.PR     025416      4933*   4935*   4937*   4938*   4943    4947    4957*   4958    4961*   4963    4967    4974*
.PRF    025324      4945    4948    4953*
.PRL    025302      4946*   4962
.PTIT   025262      4934    4939*
.PUVEC  025576      4983*   4984*   4992*   5014*
.QQ.  = 000060      5088*   5101    5102*   5103*   5104*   5105*   5106*   5107*   5108*   5109*   5110*   5111*   5112*
                    5113*   5114*   5115*   5116*   5117*   5118*   5119*   5120*   5121*   5122*   5123*   5124*   5125*
.RBIT   023274      4603*   5119
.RDLIN  025734      5062*   5106
.RDOCT  025602      5020*   5105
.SAVR6  025574      4991*   4995    5013*
.SCOPE  024674      4858*   5101
.SPASS  026356      5121    5171*
.SUSWR  027274      5123    5277*
.SVLAD  025020      4870    4875    4881*   4889
.SWOPT= 163400       760*    767     768     769     770     771     772     773     774     775     776     4341    4342
                    4344    4858    4902    4907    4920
.TBIT   021650      4351*
.TRAP   026076       973    5095*
```

```
.TRP    026114          5098    5099#
.TYPE   024536          4821#   5102
.TYPEB  025230          4933#
.TYPEO  025240          4935#   5103
.TYPES  025250          4937#   5104
.XBIT   022266          4429#   5114
```

DZRSDC.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
END          1#    760#
NEWTST       1#    760#  1078  1142  1192  1213  1222  1251  1267  1279  1288  1304  1318  1325  1341
          1355  1362  1375  1381  1387  1394  1404  1411  1419  1427  1443  1455  1467  1479  1506
          1533  1579  1615  1634  1812  1986  2074  2115  2170  2220  2327  2380  2432  2482  2529
          2602  2842  3084  3281  3492  3695  3764  3814  3866  3908  4153  4261
ODT11        1#
ODT11X       1#
POP          1#   827#  4869  4872  4968  4999  5047
PRINT        1#   827#   984   990  1007  1031  1044  1066  2021  2043  2052  2134  2147  2155  2162
          2359  2410  2464  2511  2599  3078  3687  3846  3860  4309  4326  4343  4356  4369  4536
          4909  4915  5007  5052  5072  5076  5180  5183  5189  5193  5197  5204  5209  5214  5219
          5224  5231  5236  5241  5246  5252  5272  5312  5314  5317
PUSH         1#   827#  4866  4939  4985  5022
SCOPE.       1#  1078#  1142#  1192#  1213#  1222#  1251#  1267#  1279#  1288#  1304#  1318#  1325#  1341#  1355#
          1362#  1375#  1381#  1387#  1394#  1404#  1411#  1419#  1427#  1443#  1455#  1467#  1479#  1506#  1533#
          1579#  1615#  1634#  1812#  1986#  2074#  2115#  2170#  2220#  2327#  2380#  2432#  2482#  2529#  2602#
          2842#  3084#  3281#  3492#  3695#  3764#  3814#  3866#  3908#  4153#  4261#
SET       5088#  5101  5102  5103  5104  5105  5106  5107  5108  5109  5110  5111  5112  5113  5114
          5115  5116  5117  5118  5119  5120  5121  5122  5123  5124
TYPADR       1#
TYPBIT       1#   827#
TYPCHR       1#   827#
TYPDEC       1#   827#
TYPLIN       1#   827#
TYPOCS       1#   827#  1045  1063  2023  3079  4370  4374  5258  5268
TYPOCT       1#   827#  4913  5181  5184  5190  5194  5198  5205  5215  5220  5225  5232  5237
          5242  5247  5315
TYPTXT       1#   827#   983   989  1007  1031  1044  1066  2021  2052  2134  2147  2155  2162
          2359  2410  2464  2511  2599  3078  3687  3846  3860  4326  4355  4369  4536  5180
          5183  5188  5192  5196  5204  5209  5214  5219  5224  5236  5241  5246  5252  5272
          5312  5313  5317
$CATCH       1#   760#   776
$CMTAG       1#   760#   830
$DONE        1#   760#  4336
$EQUAT       1#   760#   798
$HLT         1#   760#  4894
$KBDIN       1#   760#  5301
$KMMR        1#
$OCTAL       1#   760#  4927
$POWER       1#   760#  4975
$RAND2       1#
$RAND4       1#
$RDDEC       1#
$RDLIN       1#   760#  5055
$RDOCT       1#   760#  5015
$SCOPE       1#   760#  4851
$SET      5088#  5101  5102  5103  5104  5105  5106  5107  5108  5109  5110  5111  5112  5113  5114
          5115  5116  5117  5118  5119  5120  5121  5122  5123  5124
$SETUP       1#   760#   968
$SMMR        1#
$SUSWR       1#   760#  5276
$SWDOC       1#   760#   764
$TRAP        1#   760#  5088
$TYPE        1#   760#  4811
$TYPEA       1#
$TYPED       1#
```

SUMMR 1#

DZRSDC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADC     4340  4400  4641
ADD     1760  1762  1792  1794  1936  2057  2781  2990  3221  3430  3627  4092  4339  4399  4405
        4408  4417  4420  4448  4499  4522  4532  4605  4640  4671  4798  4842  4845  5098  5142
        5149  5168
ASL     1053
BCC     1278  1316  1353
BCS     1054  1587  1593  1603  1612  2558  2637  2876  3117  3488  3691  3786  3942
BEQ      981   987  1000  1010  1023  1034  1037  1043  1096  1100  1105  1110  1116  1122  1128
        1134  1140  1159  1163  1168  1174  1180  1185  1190  1203  1207  1211  1219  1229  1233
        1239  1243  1248  1256  1260  1264  1275  1286  1294  1298  1302  1313  1323  1331  1335
        1339  1350  1360  1368  1372  1379  1385  1392  1402  1409  1417  1425  1436  1440  1452
        1460  1464  1472  1476  1492  1495  1499  1504  1519  1526  1531  1546  1551  1554  1557
        1566  1577  1600  1608  1631  1664  1696  1759  1764  1785  1787  1798  1810  1848  1855
        1876  1887  1899  1911  1919  1923  1925  1941  1951  1963  1974  1981  2020  2033  2042
        2050  2059  2061  2063  2065  2067  2069  2071  2092  2096  2107  2111  2133  2146  2154
        2161  2191  2211  2215  2218  2318  2322  2325  2357  2365  2370  2374  2378  2408  2416
        2421  2425  2429  2462  2470  2476  2480  2510  2517  2523  2527  2570  2574  2579  2583
        2590  2834  2837  3041  3044  3047  3060  3062  3066  3273  3276  3279  3482  3485  3679
        3682  3685  3719  3731  3737  3741  3746  3760  3790  3797  3803  3810  3845  3850  3854
        3859  3864  3886  3891  3895  3901  3905  4141  4144  4147  4150  4236  4239  4244  4247
        4325  4345  4430  4507  4512  4611  4644  4649  4651  4664  4681  4684  4732  4745  4765
        4773  4779  4786  4792  4797  4828  4860  4862  4877  4889  4904  4956  4959  5031  5065
        5140  5166  5203  5208  5213  5218  5223  5230  5235  5240  5245  5251  5283  5321
BGT     4879  5033
BHIS    4710
BIC      793   978   979  1032  1049  1400  1415  1423  1502  1788  1827  1933  1934  1938  1949
        2002  2029  2036  2046  2054  2081  2099  2114  2130  2152  2169  2337  2442  2618  2752
        2777  2855  3054  3096  3298  3505  3776  3795  3801  3807  3877  3925  4058  4083  4276
        4436  4510  4515  4638  4647  4652  4658  4705  4713  4715  4720  4723  4763  4777  4790
        4807  4808  4846  5228  5309
BICB    5069
BIS      790   988  1075  1446  1449  1490  1516  1543  1649  1791  1826  1931  1955  2025  2034
        2044  2051  2103  2167  2183  2232  2244  2289  2567  2600  2617  2665  2753  2771  2776
        2854  2914  2984  3095  3145  3215  3297  3339  3358  3428  3504  3537  3555  3625  3729
        3758  3791  3798  3804  3882  3924  3971  4059  4077  4082  4279  4294  4318  4368  4483
        4492  4494  4508  4513  4517  4636  4645  4646  4655  4657  4662  4667  4682  4685  4702
        4766  4774  4780  4787  4793  4809  5249  5295
BISB    4960
BIT      980  1029  1033  1040  1042  1051  1057  1061  1127  1162  1179  1184  1408  1498  1556
        1568  1786  1924  1929  2003  2087  2137  2165  2554  2582  2597  2764  2766  2839  2982
        3063  3076  3213  3304  3426  3511  3623  3718  3783  4070  4072  4268  4324  4341  4359
        4366  4429  4438  4530  4608  4610  4643  4648  4653  4663  4692  4694  4731  4744  4752
        4772  4785  4824  4859  4863  4874  4903  4907  4920  4947  5165  5200  5202  5207  5212
        5217  5222  5229  5234  5239  5244  5250  5277
BLOS     994  4697  4700  4755
BLT     5035
BMI     4284  4299  5137
BNE     1030  1039  1041  1052  1058  1062  1521  1541  1569  1626  1683  1712  1725  1739  1775
        1802  1806  1858  1890  1914  1930  1944  1966  2004  2088  2138  2166  2204  2254  2267
        2279  2299  2555  2598  2645  2675  2700  2726  2765  2767  2769  2774  2787  2808  2840
        2883  2898  2924  2948  2981  2983  3027  3064  3077  3124  3155  3180  3212  3214  3258
        3305  3325  3331  3342  3368  3392  3425  3427  3467  3512  3524  3530  3540  3565  3589
        3622  3624  3664  3726  3784  3949  3981  4006  4032  4071  4073  4075  4080  4098  4116
        4179  4193  4195  4224  4269  4275  4286  4301  4316  4342  4360  4367  4407  4419  4439
        4447  4477  4480  4489  4491  4498  4521  4531  4604  4609  4654  4660  4666  4670  4693
        4695  4704  4712  4722  4753  4795  4825  4830  4836  4844  4864  4875  4908  4921  4948
```

	4962	4964	4998	5071	5075	5082	5154	5164	5179	5201	5271	5278	5289	5291	5303
	5305	5311													
BPL	2831	4289	4304	4323	4330	4834	4839	4918	5067	5307					
BR	794	1002	1025	1050	1056	1069	1567	1588	1604	1761	1790	1793	1932	1935	2140
	2559	2595	2639	2878	2995	3082	3119	3226	3435	3632	3812	3944	4135	4291	4328
	4450	4467	4509	4514	4534	4613	4633	4656	4676	4698	4714	4759	4800	4840	4865
	4870	4873	4934	4936	4945	5011	5045	5054	5073	5078	5141	5186	5284		
CLC	1271	1309	1346	1583	1596	3689	4749	5151	5257						
CLR	975	976	996	1020	1060	1217	1246	1284	1321	1358	1390	1407	1512	1523	1537
	1548	1562	1701	1828	1829	1979	2048	2401	2504	2551	2563	2632	2634	2862	2871
	2873	2896	2968	3112	3114	3199	3323	3329	3340	3412	3522	3528	3538	3609	3937
	3939	4208	4271	4357	4358	4437	4440	4441	4451	4454	4461	4464	4475	4607	4614
	4615	4618	4621	4627	4630	4679	4736	4748	4756	4760	4771	4784	4937	4996	5026
	5027	5128	5129	5135	5173	5174	5261								
CLRB	1447	4922	4944	4946	4966										
CMP	993	1009	1036	1038	1095	1099	1104	1133	1158	1189	1202	1206	1210	1218	1228
	1232	1238	1242	1247	1255	1259	1263	1274	1293	1297	1301	1312	1330	1334	1338
	1349	1367	1371	1378	1384	1401	1416	1424	1435	1439	1451	1459	1463	1471	1475
	1491	1494	1503	1518	1525	1530	1540	1545	1550	1553	1565	1576	1599	1607	1629
	1630	1758	1763	1784	1797	1809	1875	1886	1898	1910	1918	1922	1940	1950	1962
	1973	1980	2019	2032	2041	2049	2058	2060	2062	2064	2066	2068	2070	2095	2110
	2132	2145	2153	2160	2190	2210	2217	2317	2324	2356	2364	2369	2377	2407	2415
	2420	2428	2461	2469	2479	2509	2516	2526	2569	2573	2587	2836	3040	3046	3061
	3067	3081	3272	3278	3481	3678	3684	3730	3736	3740	3745	3759	3789	3796	3802
	3844	3849	3853	3862	3885	3890	3894	3900	4140	4143	4146	4149	4235	4238	4246
	4406	4418	4497	4520	4650	4665	4669	4696	4699	4703	4709	4711	4721	4754	4843
	4871	4963	5064	5282	5285	5288	5304								
CMPB	986	4829	4861	4878	5030	5032	5034	5070	5074	5081	5310				
DEC	1001	1022	1520	1625	1682	1711	1724	1738	1774	1801	1805	1857	1889	1913	1943
	1965	2203	2253	2266	2278	2298	2644	2674	2699	2725	2768	2773	2786	2807	2882
	2897	2923	2947	2980	3026	3059	3065	3123	3154	3179	3211	3257	3324	3330	3341
	3367	3391	3424	3466	3523	3529	3539	3564	3588	3621	3663	3723	3948	3980	4005
	4031	4074	4079	4097	4115	4178	4192	4194	4223	4274	4285	4300	4315	4457	4624
	4659	4794	4796	4835	5153	5163	5270								
EMT	799														
HALT	777	1008	1068	4919	4993	5010									
INC	1024	1055	1538	1549	1563	1895	1920	2556	4691	4906	4997	5044	5138		
INCB	4881	4886	4957	4961											
JMP	787	791	982	1016	1070	1628	1952	1956	2072	2168	2841	3490	3693	4270	4310
	4335	4349	4361	4362	4924	5008	5187	5227							
JSR	2093	2097	2108	2112	2148	2156	2163	2778	2986	3217	3489	3688	3692	4089	4347
	4495	4518	4535	4634	4674	4762	4776	4789	4916						
MOV	968	969	970	971	972	973	974	992	995	1003	1004	1005	1006	1011	1012
	1019	1021	1026	1027	1028	1035	1046	1059	1063	1067	1076	1077	1082	1083	1084
	1085	1086	1087	1088	1089	1090	1098	1147	1148	1149	1150	1151	1152	1153	1154
	1155	1156	1157	1161	1200	1201	1205	1209	1216	1231	1234	1235	1237	1241	1245
	1250	1254	1258	1262	1270	1272	1273	1283	1292	1296	1300	1308	1310	1311	1320
	1329	1333	1337	1345	1347	1348	1357	1366	1377	1383	1389	1398	1399	1406	1413
	1414	1421	1422	1433	1445	1448	1450	1457	1469	1487	1488	1489	1497	1501	1513
	1514	1515	1517	1524	1528	1529	1539	1542	1544	1552	1564	1574	1575	1584	1585
	1589	1591	1597	1598	1605	1606	1620	1621	1622	1623	1624	1673	1702	1715	1729
	1771	1772	1795	1796	1803	1807	1808	1842	1873	1874	1882	1884	1885	1896	1897
	1905	1906	1908	1909	1921	1939	1948	1953	1954	1958	1960	1961	1971	1972	1978
	2005	2016	2017	2018	2022	2023	2030	2031	2039	2040	2047	2089	2090	2104	2105
	2128	2129	2131	2136	2139	2141	2142	2143	2144	2150	2151	2158	2159	2189	2192
	2193	2198	2213	2243	2264	2268	2273	2288	2320	2342	2343	2350	2354	2355	2393

```
            2394  2405  2406  2447  2448  2455  2459  2460  2496  2497  2507  2508  2548  2550  2552
            2560  2561  2562  2566  2586  2592  2593  2615  2631  2633  2638  2640  2642  2643  2648
            2649  2650  2664  2690  2716  2745  2746  2747  2772  2780  2782  2783  2785  2798  2670
            2872  2877  2879  2880  2881  2891  2892  2893  2894  2895  2913  2938  2956  2957  2961
            2962  2969  2971  2972  2985  2988  2989  2991  2992  2994  3017  3055  3056  3057  3058
            3068  3069  3070  3071  3073  3074  3075  3079  3103  3111  3113  3118  3120  3121  3122
            3128  3129  3130  3144  3170  3187  3188  3192  3193  3200  3202  3203  3216  3219  3220
            3222  3223  3225  3248  3295  3306  3321  3322  3326  3327  3328  3334  3335  3336  3337
            3338  3357  3382  3400  3401  3405  3406  3413  3415  3416  3429  3431  3432  3434  3456
            3502  3513  3520  3521  3525  3526  3527  3532  3533  3534  3535  3536  3554  3579  3597
            3598  3602  3603  3610  3612  3613  3626  3628  3629  3631  3653  3712  3713  3715  3716
            3717  3722  3727  3728  3747  3748  3756  3757  3777  3778  3787  3788  3827  3828  3835
            3842  3843  3883  3920  3922  3923  3936  3938  3943  3945  3946  3947  3952  3953  3954
            3955  3956  3970  3996  4022  4051  4052  4053  4078  4088  4091  4093  4094  4096  4106
            4164  4170  4171  4172  4175  4176  4177  4181  4186  4187  4188  4189  4190  4191  4214
            4273  4277  4278  4280  4281  4282  4293  4295  4296  4297  4314  4317  4319  4320  4321
            4344  4370  4374  4397  4398  4403  4404  4412  4413  4415  4416  4431  4432  4442  4444
            4445  4449  4493  4496  4516  4519  4528  4529  4606  4612  4616  4635  4637  4639  4642
            4661  4668  4675  4730  4733  4734  4735  4743  4746  4747  4761  4767  4775  4788  4799
            4801  4806  4821  4822  4823  4826  4841  4847  4848  4849  4866  4867  4869  4872  4882
            4883  4887  4890  4910  4913  4933  4940  4941  4942  4943  4968  4969  4970  4971  4983
            4984  4985  4986  4987  4988  4989  4990  4991  4992  4995  4999  5000  5001  5002  5003
            5004  5005  5006  5020  5021  5022  5023  5024  5028  5046  5047  5048  5049  5062  5063
            5084  5095  5097  5099  5126  5127  5132  5148  5150  5160  5181  5184  5190  5194  5198
            5205  5210  5215  5220  5225  5232  5237  5242  5247  5253  5254  5258  5260  5268  5279
            5280  5281  5286  5287  5293  5294  5308  5315  5319  5322
MOVB        1370  1434  1438  1458  1462  1470  1474  2581  4831  4832  4837  4880  4912  4935  4938
            4965  5029  5068  5079
NOP         3724  3725  4133  4152  4333
RESET       1091  1227  4346
ROL          998  1277  1315  1352  1586  1592  1602  1611  2557  2636  2875  3116  3487  3690  3785
            3941  4452  4453  4455  4456  4462  4463  4465  4466  4533  4619  4620  4622  4623  4628
            4629  4631  4632  4750  4751  4757  4758  4949  4951  4953  5039  5040  5041  5042  5043
            5152  5255  5256  5262  5263  5264  5265  5266  5267
ROLB        4950  4952  4954
ROR         5036  5037  5038
RTI         4395  4401  4409  4414  4421  4443  4458  4500  4523  4617  4625  4672  4850  4884  4891
            4923  4972  5050  5085  5130  5133  5167  5169  5175  5296  5323
RTS         4372  4376  4537  4541  4543  4545  4547  4549  4551  4553  4555  4557  4559  4561  4563
            4565  4567  4569  4571  4686  4706  4716  4724  4802  4810  5143  5155  5273
SEC          997  1590  1610  2635  2874  3115  3940
SUB         1789  1937  2784  2970  2993  3201  3224  3414  3433  3611  3630  4095  4911  5096
TRAP        5098  5101  5102  5103  5104  5105  5106  5107  5108  5109  5110  5111  5112  5113  5114
            5115  5116  5117  5118  5119  5120  5121  5122  5123  5124
TST          999  1109  1115  1121  1139  1167  1173  1285  1322  1359  1391  1663  1695  1847  1854
            2091  2106  2214  2321  2373  2424  2475  2522  2553  2578  2830  2833  3043  3275  3484
            3681  3809  3858  3904  4243  4288  4303  4329  4446  4476  4479  4487  4490  4506  4511
            4603  4680  4683  4764  4778  4791  4868  4888  4917  5139  5178  5290  5302  5320
TSTB        4283  4298  4322  4827  4833  4838  4876  4955  4958  5066  5136  5306
.ASCIZ       985   991  1008  1032  1045  1067  2022  2044  2053  2135  2148  2156  2163  2360  2411
            2465  2512  2600  3079  3688  3847  3861  4310  4327  4344  4357  4370  4378  4384  4537
            4572  4573  4574  4575  4577  4579  4581  4583  4585  4587  4589  4591  4593  4594  4595
            4597  4910  4916  5008  5054  5073  5078  5181  5184  5190  5194  5198  5205  5210  5215
            5220  5225  5232  5237  5242  5247  5253  5273  5313  5315  5318
.BLKB        5087
.BLKW       4974  5327  5328
```

DZRSDC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
.ENABL      1   760
.END     5329
.ENDC     827   969   975  1081  1145  1195  1213  1225  1251  1267  1279  1288  1304  1318  1325
         1341  1355  1362  1375  1381  1387  1394  1404  1411  1419  1430  1443  1455  1467  1482
         1509  1533  1579  1618  1637  1815  1989  2077  2118  2173  2223  2330  2383  2435  2485
         2532  2605  2845  3087  3284  3495  3698  3767  3817  3869  3911  4156  4264  4341  4343
         4344  4350  4864  4892  4907  4909  4917  4925  4991  4999  5008  5009  5065  5088  5102
         5103  5104  5105  5106  5107  5108  5109  5110  5111  5112  5113  5114  5115  5116  5117
         5118  5119  5120  5121  5122  5123  5124  5125
.EVEN     985   991  1008  1032  1045  1067  2022  2044  2053  2135  2148  2156  2163  2360  2411
         2465  2512  2600  3079  3688  3847  3861  4310  4327  4344  4357  4370  4391  4537  4910
         4916  5008  5054  5073  5078  5181  5184  5190  5194  5198  5205  5210  5215  5220  5225
         5232  5237  5242  5247  5253  5273  5313  5315  5318
.IF       811   968   975  1078  1142  1192  1213  1222  1251  1267  1279  1288  1304  1318  1325
         1341  1355  1362  1375  1381  1387  1394  1404  1411  1419  1427  1443  1455  1467  1479
         1506  1533  1579  1615  1634  1812  1986  2074  2115  2170  2220  2327  2380  2432  2482
         2529  2602  2842  3084  3281  3492  3695  3764  3814  3866  3908  4153  4261  4341  4342
         4344  4349  4858  4892  4902  4907  4916  4920  4991  4999  5007  5008  5064  5087  5101
         5102  5103  5104  5105  5106  5107  5108  5109  5110  5111  5112  5113  5114  5115  5116
         5117  5118  5119  5120  5121  5122  5123  5124
.IFF      969  4342  4349  4864  4902  4920  5008  5065  5088  5101  5102  5103  5104  5105  5106
         5107  5108  5109  5110  5111  5112  5113  5114  5115  5116  5117  5118  5119  5120  5121
         5122  5123  5124
.IIF      767   768   769   770   771   772   773   774   775   776  4351  4884  4888  4893  4926
         4927  5087
.IRP     4866  4869  4872  4940  4968  4985  4999  5022  5047  5101
.LIST       1   760   777   827   985   991  1008  1032  1045  1067  1078  1082  1142  1146  1192
         1196  1213  1214  1222  1226  1251  1252  1267  1268  1279  1280  1288  1289  1304  1305
         1318  1319  1325  1326  1341  1342  1355  1356  1362  1363  1375  1376  1381  1382  1387
         1388  1394  1395  1404  1405  1411  1412  1419  1420  1427  1431  1443  1444  1455  1456
         1467  1468  1479  1483  1506  1510  1533  1534  1579  1580  1615  1619  1634  1638  1812
         1816  1986  1990  2022  2044  2053  2074  2078  2115  2119  2135  2148  2156  2163  2170
         2174  2220  2224  2327  2331  2360  2380  2384  2411  2432  2436  2465  2482  2486  2512
         2529  2533  2600  2602  2606  2842  2846  3079  3084  3088  3281  3285  3492  3496  3688
         3695  3699  3764  3768  3814  3818  3847  3861  3866  3870  3908  3912  4153  4157  4261
         4265  4310  4327  4344  4357  4370  4537  4910  4916  5008  5054  5073  5078  5088  5101
         5102  5103  5104  5105  5106  5107  5108  5109  5110  5111  5112  5113  5114  5115  5116
         5117  5118  5119  5120  5121  5122  5123  5124  5125  5181  5184  5190  5194  5198  5205
         5210  5215  5220  5225  5232  5237  5242  5247  5253  5273  5313  5315  5318
.MACRO       1  5088
.MCALL     760   827  1078  1142  1192  1213  1222  1251  1267  1279  1288  1304  1318  1325  1341
         1355  1362  1375  1381  1387  1394  1404  1411  1419  1427  1443  1455  1467  1479  1506
         1533  1579  1615  1634  1812  1986  2074  2115  2170  2220  2327  2380  2432  2482  2529
         2602  2842  3084  3281  3492  3695  3764  3814  3866  3908  4153  4261
.NLIST      1   760   777   827   985   991  1008  1032  1045  1067  1078  1082  1142  1146  1192
         1196  1213  1214  1222  1226  1251  1252  1267  1268  1279  1280  1288  1289  1304  1305
         1318  1319  1325  1326  1341  1342  1355  1356  1362  1363  1375  1376  1381  1382  1387
         1388  1394  1395  1404  1405  1411  1412  1419  1420  1427  1431  1443  1444  1455  1456
         1467  1468  1479  1483  1506  1510  1533  1534  1579  1580  1615  1619  1634  1638  1812
         1816  1986  1990  2022  2044  2053  2074  2078  2115  2119  2135  2148  2156  2163  2170
         2174  2220  2224  2327  2331  2360  2380  2384  2411  2432  2436  2465  2482  2486  2512
         2529  2533  2600  2602  2606  2842  2846  3079  3084  3088  3281  3285  3492  3496  3688
         3695  3699  3764  3768  3814  3818  3847  3861  3866  3870  3908  3912  4153  4157  4261
         4265  4310  4327  4344  4357  4370  4537  4910  4916  5008  5054  5073  5078  5088  5101
         5102  5103  5104  5105  5106  5107  5108  5109  5110  5111  5112  5113  5114  5115  5116
         5117  5118  5119  5120  5121  5122  5123  5124  5125  5181  5184  5190  5194  5198  5205
```

F12

DZRSDC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
          5210    5215    5220    5225    5232    5237    5242    5247    5253    5273    5313    5315    5318
.PAGE      797     830    4851    4894    4927    4975    5015    5055    5088
.REM         1
.REPT      777
.SBTTL    1078    1142    1192    1222    1427    1479    1506    1615    1634    1812    1986    2074    2115    2170    2220
          2327    2380    2432    2482    2529    2602    2842    3084    3281    3492    3695    3764    3814    3866    3908
          4153    4261    4336    4811    4851    4894    4927    4975    5015    5055    5088
.TITLE     760
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DZRSDC.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZRSDC.SML,DZRSDC.P11
RUN-TIME: 30 50 8 SECONDS
RUN-TIME RATIO: 137/89=1.5
CORE USED: 23K (45 PAGES)

Spooler runtime 19 Seconds. 82 KCS, 543 disk reads, 4 disk writes, 148 pages