RH70-RS04

MAINT MODE DIAGNOSTIC

MD-11-DERSD-C

EP-DERSD-C-DL-B

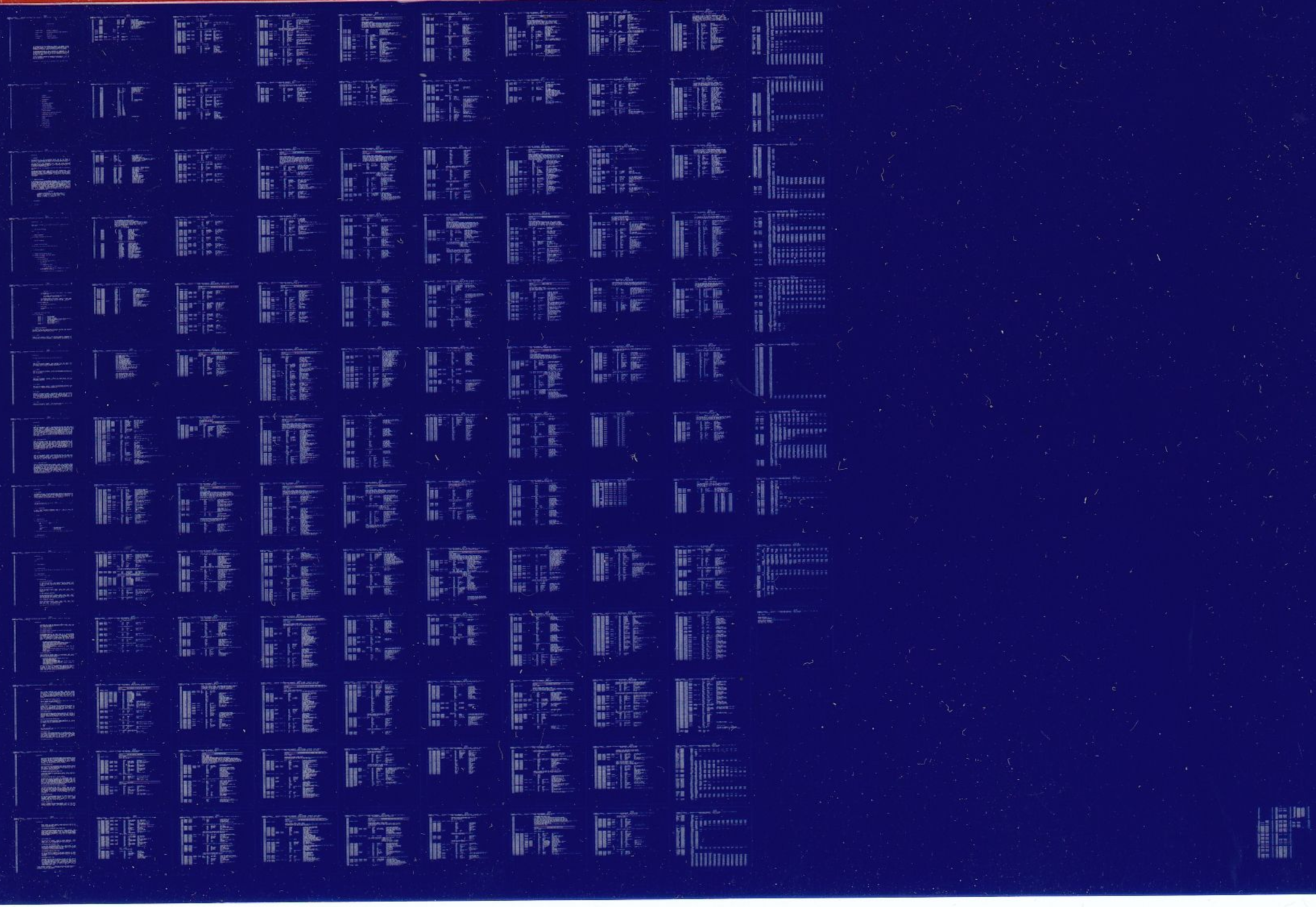
COPYRIGHT © 1977

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



```
                        .REM !


                                        IDENTIFICATION

                                        --------------



                           PRODUCT CODE:        MAINDEC-11-DERSD-C-D

                           PRODUCT NAME:        RH70-RS04 MAINTENANCE MODE DIAGNOSTIC

                           DATE:                MARCH-1977

                           MAINTAINER:          DIAGNOSTIC GROUP

                           AUTHORS:             STANLEY HARACKIEWICZ








                        THE INFORMATION IN THIS DOCUMENT IS  SUBJECT  TO  CHANGE  WITHOUT
                        NOTICE  AND  SHOULD  NOT  BE CONSTRUED AS A COMMITMENT BY DIGITAL
                        EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT CORPORATION ASSUMES  NO
                        RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

                        THE SOFTWARE DESCRIBED IN  THIS  DOCUMENT  IS  FURNISHED  TO  THE
                        PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND
                        CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY
                        FOR  USE  IN  SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN
                        WRITING BY DIGITAL.

                        DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY  FOR  THE
                        USE  OR  RELIABILITY  OF  ITS  SOFTWARE  ON EQUIPMENT THAT IS NOT
                        SUPPLIED BY DIGITAL.


                        COPYRIGHT (C) 1975,1977 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
```

TABLE OF CONTENTS

CONTENTS

DESCRIPTION

1. ABSTRACT

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

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

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

THE 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

DESCRIPTION

PDP-11 WITH A MINIMUM OF 8K OF MEMORY AND AN RH11 CONTROLLER WITH A RS04 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/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

STARTING ADDRESSES

1. STARTING ADDRESS 200

   A. SET SWITCHES (SEE SECTION 5)

   B. PRESS START

   C. THE PROGRAM WILL TYPE:

      TEST ALL DRIVES? (Y OR N)

   D. IF THE OPERATOR TYPES "Y" THE PROGRAM WILL TEST ALL RS04 DRIVES ON THE SYSTEM

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

```
DESCRIPTION


                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

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
```

DESCRIPTION

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 REGISTER TO EQUAL THE VALUE FOLLOWING THE MRCK INSTRUCTION. IF THE MAINTENANCE REGISTER DOES NOT COMPARE, THE PROGRAM RETURNS TO THE "HLT" INSTRUCTION FOLLOWING THE CORRECT VALUE AND PRINTS OUT THE ERROR. IF THE MAINTENANCE REGISTER IS CORRECT, THE PROGRAM RETURNS TO THE INSTRUCTION FOLLOWING THE "HLT" INSTRUCTION.

5.1.7 DSCK

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

DESCRIPTION

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 MWD8 (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 MWD8 (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.

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 MRD8 (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 NCT 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.

DESCRIPTION

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.
         GOOD                    = EXPECTED DATA.
         BAD                     = DATA RECEIVED.
```

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

6.2 ERROR RECOVERY

RESTART AT 200 OR AT 220

```
DESCRIPTION


7.  RESTRICTIONS

NONE



8.  MISCELLANEOUS



8.1  EXECUTION TIME

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



8.2  STACK POINTER

STACK IS INITALLY SET TO 500



9.  TEST DESCRIPTION

     1.  TEST FOR ONLINE DRIVES

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

     2.  RESET TEST FOR REGISTERS

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

     3.  SET AND CLEAR ALL REGISTERS

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

     4.  TEST "CLEAR BIT" IN RSCS2

         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.  LOAD RSDB WITH ALL ONES AND ALL ZEROS
```

```
                        DESCRIPTION

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

                              6.  TEST PROGRAM INTERRUPT

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

                              7.  MAINTENANCE TIMING TEST

                                  THE FOLLOWING TEST ON  THE  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.

                              8.  SECTOR FRACTION TEST

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

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

                                  WHEN THE LAST WORD IS BEING TRANSFERRED, SECTOR AND  FRACTION
                                  IS  EQUAL  TO  7777  TO  INDICATE  LAST WORD ON THIS TRACK --
                                  HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER  WILL
                                  CLEAR  THE  FRACTION  BITS  IF ANOTHER WORD IS CLOCKED.  RSLA
                                  SHOULD INDICATE 7700 ON ANOTHER MAINTENANCE CLOCK.

                              9.  DISK ILLEGAL FUNCTION TEST
```

DESCRIPTION

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

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

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

11. DRIVE SEARCH TEST 1

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

12. DRIVE SEARCH TEST 2

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

13. REGISTER MODIFICATION REFUSED TEST

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

1. RSCS1
2. RSDA
3. RSER

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

14. DRIVE SELECT TEST

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

15. MAINTENANCE WRITE TEST

DESCRIPTION

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.

16. 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.)

17. 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.

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

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.

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

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

20. IGNORE FUNCTION TEST

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

21. INVALID ADDRESS TEST

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

```

DESCRIPTION

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

        22.     DISK OPERATION INCOMPLETE (OPI) ERROR TEST

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



        23.     PARITY ERROR TEST

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

        24.     MAINTENANCE MODE INTERRUPT TEST

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

        25.     DISK ADDRESS OVERFLOW (AOE) TEST

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

        26.     MAINTENANCE VERIFY TEST

                THIS TEST WILL ONLY RUN IF SWITCH 11 IS SET IN THE "SWITCH
                REGISTER" FOR IT WILL ACTUALLY WRITE DATA ONTO THE DISK.  IT
                WILL WRITE ONE TRACK OF ALL ONES.  THE DRIVE IS THEN PLACED
                IN MAINTENANCE MODE AND IT WILL THEN WRITE ONE SECTOR OF THE
                SAME TRACK WITH ALL ZEROS.  THE DRIVE IS THEN TAKEN OUT OF
                "MAINTENANCE MODE" AND THE TRACK IS THEN READ.  THE TRACK
                SHOULD CONTAIN ALL ONES.
!
.TITLE  MAINDEC-11-DERSD-C      RH70-RS04 MAINTENANCE MODE DIAGNOSTIC
;COPYRIGHT 1974,1975,1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
;PROGRAM BY STANLEY MARACKIEWICZ


;          SWITCH                   USE
;          ------            -------------------
;
```

```
        100000                                  SW15=   100000          ;HALT ON ERROR
        040000                                  SW14=   40000           ;LOOP ON TEST
        020000                                  SW13=   20000           ;INHIBIT ERROR TYPEOUTS
        010000                                  SW12=   10000           ;TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
        004000                                  SW11=   4000            ;RUN MAINTENANCE MODE VERIFY TEST
        002000                                  SW10=   2000            ;0 - BELL ON PASS COMPLETE
                                                                        ;1 - BELL ON ERROR
        001000                                  SW9=    1000            ;LOOP ON ERROR
        000400                                  SW8=    400             ;LOOP ON TEST IN SW<7:0>
        000000                          .=      0                       ;TRAP CATCHER FROM 0 - 776
        000046                          .=      46                      ;HOOKS FOR ACT 11
000046  021200                          $ENDAD
        000052                          .=      52
000052  040000                          BIT14
        000200                          .=      200
000200  000137  000232                          JMP     @#BEGIN1

        000220                          .=      220
000220  052767  000100  000716                  BIS     #BIT6,FLAG2     ;TEST ALL DRIVES
000226  000137  001234          BEGIN2: JMP     @#BEGIN

000232  042767  000100  000704  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
        177570                          SWR=      177570                ;SWITCH REGISTER
        177570                          DISPLAY=SWR                     ;DISPLAY REGISTER
        000007                          BELL=     7                     ;BELL
        000000                          R0=       %0                    ;R0 - DEFINE REGISTERS
        000001                          R1=       %1                    ;R1
        000002                          R2=       %2                    ;R2
        000003                          R3=       %3                    ;R3
        000004                          R4=       %4                    ;R4
        000005                          R5=       %5                    ;R5
        000006                          SP=       %6                    ;R6 - STACK POINTER
        000007                          PC=       %7                    ;R7 - PROGRAM COUNTER
        000001                          BIT0=     1                     ;BIT EQUATES
        000002                          BIT1=     2
        000004                          BIT2=     4
        000010                          BIT3=     10
        000020                          BIT4=     20
        000040                          BIT5=     40
        000100                          BIT6=     100
        000200                          BIT7=     200
        000400                          BIT8=     400
        001000                          BIT9=     1000
        002000                          BIT10=    2000
        004000                          BIT11=    4000
        010000                          BIT12=    10000
        020000                          BIT13=    20000
        040000                          BIT14=    40000
        100000                          BIT15=    100000
        000001                  GOOD=   %1                              ;FOR GOOD DATA
        000000                  BAD=    %0                              ;FOR BAD DATA

```

```
          001000                     .=       1000

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    177566                     TPB:     177566              ;OUTPUT BUFFER

          001100                     .=       1100

                                     ;DISK I/O REGISTERS

001100    172040                     RSCS1:   172040              ;DISK CONTROL + STATUS REGISTER
001102    172050                     RSCS2:   172050              ;DISK CONTROL + STATUS REGISTER
001104    172042                     RSWC:    172042              ;WORD COUNT REGISTER
001106    172044                     RSBA:    172044              ;BUS ADDRESS
001110    172046                     RSDA:    172046              ;DISK ADDRESS (DESIRED ADDRESS)
001112    172052                     RSDS:    172052              ;DRIVE STATUS
001114    172054                     RSER:    172054              ;ERROR REG.
001116    172056                     RSAS:    172056              ;ATTENTION SUMMARY
001120    172060                     RSLA:    172060              ;LOOK AHEAD
001122    172062                     RSDB:    172062              ;DATA BUFFER REGISTER
001124    172064                     RSMR:    172064              ;MAINTENANCE REGISTER
001126    172066                     RSDT:    172066              ;DRIVE TYPE REGISTER
001130    000204                     RSVEC:   204                 ;INTERRUPT VECTOR
001132    000206                     RSVCPS:  206                 ;INTERRUPT PRIO. VECTOR
001134    172041                     RSCS1B:  172041              ;ODD BYTE ADD FOR CS1
001136    172051                     RSCS2B:  172051              ;ODD BYTE ADD FOR CS2
001140    172043                     RSWCB:   172043              ;ODD BYTE ADD FOR CW
001142    172045                     RSBAB:   172045              ;ODD BYTE ADD FOR BA

```

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

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

                          ;BIT ASSIGNMENTS FOR THE REGISTER BITS

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

```
    835                                     ;WORKING LOCATIONS
    836
    837  001144  000000                     FLAG2:  0                             ;SECOND FLAG WORD
    838  001146  000000                     LSTEV:  0                             ;LAST EVEN BIT TRANSFERED
    839  001150  000000                     LSTOD:  0                             ;LAST ODD BIT TRANSFERED
    840  001152  000000                     NOWEV:  0                             ;PRESENT EVEN BIT BEING XFERED
    841  001154  000000                     NOWOD:  0                             ;PRESENT ODD BIT BEING XFERED
    842  001156  000000                     RSD:    0                             ;SAME
    843  001160  000000                     UNNUM:  0                             ;UNIT CURRENTLY BEING TESTED
    844  001162  000000                     UNITSV: 0                             ;SET BIT=UNIT ON BUS
    845  001164  000000                     UNCMP:  0                             ;FOR COMPARING FOR # OF DEVICE
    846  001166  000000                     ONCEE:  0                             ;DID WE TEST ANY DRIVES
    847  001170  000000                     TIMSV:  0                             ;SAVE LOC FOR TIME
    848          172100                     MPRO=172100                           ;PARITY REG
    849  001172  000000                     SAVEE:  0                             ;WORK LOC
    850  001174  000000  000000             MCCNT:  0,0                           ;MAINT CLOCK COUNT
    851  001200  000000                     WCRC:   0                             ;WORK LOC FOR CREATING CRC WORD
    852  001202  000000                     REPT:   0                             ;REPEAT COUNTER
    853  001204  000000                     REPT1:  0                             ;REPEAT COUNTER
    854  001206  000000                     CLKCNT: 0                             ;CLOCK COUNTER FOR EACH WORD
    855  001210  000000                     INBIT:  0                             ;USED IN CRC CAL ROUTINE
    856  001212  000000                     WK15:   0                             ;USED IN CRC CAL ROUTINE
    857  001214  000000                     WORK:   0
    858  001216  000000                     WORK0:  0
    859  001220  000000                     WORK1:  0
    860  001222  000000                     WORK2:  0
    861  001224  000000                     WORK3:  0
    862  001226  000000                     WORK4:  0
    863  001230  000000                     WORK5:  0
    864  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
```

```
001234  012706  000500          BEGIN:  MOV     #500,SP         ;SET STACK TO  *** 500 ***
001240  012737  025014  000024          MOV     #.POWER,@#24    ;SET UP PF VECTOR
001246  012737  000340  000026          MOV     #340,@#26       ;LOCK OUT THE WORLD
001254  012737  024432  000030          MOV     #.HLT,@#30      ;SET EMT VECTOR
001262  012737  000340  000032          MOV     #340,@#32       ;LOCK UP
001270  012737  025416  000034          MOV     #.TRAP,@#34     ;SET TRAP VECTOR
001276  012737  000340  000036          MOV     #340,@#36       ;LOCK UP
001304  005067  177470                  CLR     ICNT            ;INIT ICNT
001310  005067  177474                  CLR     LAD             ;INIT LAD
001314  042767  177677  177622          BIC     #177677,FLAG2
001322  042767  153777  177636          BIC     #153777,ONCEE
001330  032767  000100  177606          BIT     #BIT6,FLAG2     ;TEST ALL DRIVES?
001336  001402                          BEQ     5$              ;ASK
001340  000137  001672                  JMP     @#MULTII
001344                          5$:
001344  104402  001350                  TYPE    ,.+2            ;.ASCIZ <15><12>"TEST ALL DRIVES? (Y OR N) "
001406  104412                          RDLIN
001410  122767  000131  023760          CMPB    #'Y,INPUT       ;TEST FOR YES
001416  001525                          BEQ     MULTII          ;YES
001420  052767  000020  177540          BIS     #BIT4,ONCEE     ;SET TEST ONLY ONE DRIVE FLAG
001426                          1$:
001426  104402  001432                  TYPE    ,.+2            ;.ASCIZ "TYPE UNIT #"
001446  104410                          RDOCT
001450  012604                          MOV     (6)+,R4         ;GET NUMBER
001452  022704  000010                  CMP     #10,R4          ;CORRECT #
001456  101763                          BLOS    1$              ;NO
001460  010467  177474                  MOV     R4,UNNUM        ;SET UNIT #
001464  005002                          CLR     R2              ;CLEAR WORK AREA
001466  000261                          SEC                     ;SET CARRY
001470  006102                  2$:     ROL     R2              ;SET WORK BIT
001472  005704                          TST     R4              ;IS THIS BIT CORRESPOND WITH CORRECT DRIVE #
001474  001402                          BEQ     3$              ;YES
001476  005304                          DEC     R4              ;NO TRY AGAIN
001500  000773                          BR      2$              ;TEST AGAIN
001502  010267  177454          3$:     MOV     R2,UNITSV       ;SET DRIVE BIT IN UNITSV
001506  010267  177452                  MOV     R2,UNCMP        ;SET UNIT COMPARE
001512  016777  177442  177362          MOV     UNNUM,@RSCS2    ;LOAD DRIVE
001520  012777  177777  177366          MOV     #-1,@RSER       ;LOAD ERRORS
001526  104402  001532                  TYPE    ,.+2            ;.ASCIZ "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD BE ON
001642  000000                          HALT                    ;WAIT FOR LIGHTS TO BE CHECKED
001644  026777  177312  177244          CMP     UNITSV,@RSAS    ;DID CORRECT ATA SET
001652  001405                          BEQ     4$
001654  017700  177236                  MOV     @RSAS,BAD       ;GET RSAS
001660  016701  177276                  MOV     UNITSV,GOOD     ;GET CORRECT AND
001664  104000                          HLT                     ;RSAS=BAD GOOD=CORRECTIONS
                                                                ;ATA BIT SHOULD SET FOR ERRORS
                                                                ;WERE SET IN RSER
001666  000167  000430          4$:     JMP     NOWGO           ;START TESTING
```

```
                                                ;NOW TEST FOR DRIVES

001672  012701  000010                  MULTII: MOV     #8.,R1                  ;PUT 8 INTO R1 FOR COUNT
001676  005077  177200                          CLR     @RSCS2                  ;SET DEVICE TO ZERO
001702  012777  177777  177204          TRY:    MOV     #-1,@RSER               ;CAUSE AN ERROR +SETS BIT IN RSAS REG
001710  005301                                  DEC     R1                      ;DO A MAXIMUM OF 8 TIMES
001712  001403                                  BEQ     DVNUM                   ;TESTED FOR ALL DRIVES GET OUT
001714  005277  177162                          INC     @RSCS2                  ;INCREMENT DRIVE UNIT
001720  000770                                  BR      TRY                     ;REPEAT FOR NEXT DRIVE
001722  017767  177170  177232          DVNUM:  MOV     @RSAS,UNITSV            ;SAVE
001730  012767  000401  177226                  MOV     #401,UNCMP              ;SETUP TO CMP WITH UNITSV
001736  012767  000000  177214                  MOV     #0,UNNUM                ;PUT 0 INTO UNIT NO.
001744  032767  020000  175616                  BIT     #BIT13,SWR              ;INHIBIT TYPE OUT?
001752  001015                                  BNE     STTEST                  ;YES
001754  104402  001760                          TYPE    ,.+2                    ;.ASCIZ <15><12>"TESTING UNIT "
002000  042767  100000  177160                  BIC     #BIT15,ONCEE            ;CLEAR ERROR FLAG
002006  036767  177152  177146          STTEST: BIT     UNCMP,UNITSV            ;IS THIS DRIVE ON THE SYSTEM
002014  001440                                  BEQ     TRYNX                   ;NO
002016  016777  177136  177056                  MOV     UNNUM,@RSCS2            ;YES PUT UNIT # INTO CS2
002024  022777  000002  177074          3$:     CMP     #2,@RSDT                ;IS THIS A RS04?
002032  001404                                  BEQ     1$                      ;YES
002034  022777  000003  177064                  CMP     #3,@RSDT                ;IS IT A RS04?
002042  001025                                  BNE     TRYNX                   ;GET A NEW NUMBER
002044  032767  020000  175516          1$:     BIT     #BIT13,SWR              ;INHIBIT TYPE OUT?
002052  001020                                  BNE     4$                      ;YES
002054  032767  100000  177104                  BIT     #BIT15,ONCEE            ;ANY ERRORS?
002062  001404                                  BEQ     5$                      ;NO
002064  104402  002070                          TYPE    ,.+2                    ;.ASCIZ <15><12><12>
002074                                  5$:
002074  016746  177060                          MOV     UNNUM,-(6)              ;PUT UNNUM ON STACK
002100  104406                                  TYPES                           ;TYPE STACK IN OCTAL - SUPRESS
002102  104402  000040                          TYPE    ,40                     ;TYPE SPACE
002106  042767  100000  177052                  BIC     #BIT15,ONCEE            ;CLEAR ERROR FLAG
002114  000502                          4$:     BR      NOWGO                   ;NOW TEST
002116  032767  000020  177042          TRYNX:  BIT     #BIT4,ONCEE             ;MULTI DRIVE
002124  001074                                  BNE     DONEE                   ;NO
002126  006367  177032                  1$:     ASL     UNCMP                   ;CHECK NEXT BIT FOR DRIVE
002132  103403                                  BCS     CHCKDV                  ;DID WE TEST ANY REG?
002134  005267  177020                          INC     UNNUM                   ;INC UNIT #
002140  000722                                  BR      STTEST                  ;CHECK FOR NEXT DRIVE
```

```
002142  032767  000001  177016  CHCKDV: BIT     #BIT0,ONCEE     ;DID WE TEST ANY DRIVES?
002150  001062                          BNE     DONEE           ;YES WE DID TEST A DRIVE
002152  012767  100000  177004          MOV     #100000,UNCMP   ;NO DRIVES TESTED, COULD NOT SET
002160  005067  176774                  CLR     UNNUM           ;ANY AS BITS, THUS DEFAULTS TO
002164  032767  020000  175376          BIT     #BIT13,SWR      ;INHIBIT TYPE OUT?
002172  001050                          BNE     4$              ;YES
002174  016746  176760                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
002200  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
002202  104402  000040                  TYPE    ,40             ;TYPE SPACE
002206  104402  002212                  TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
002304  012767  000001  176652          MOV     #1,UNCMP
002312  000000                          HALT                    ;WAIT
002314  000402                  4$:     BR      NOWGO           ;TEST DRIVE 0
002316  000167  016612          DONEE:  JMP     DONE            ;GET OUT

                                ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
                                ;TO CLEAR ALL THE RH AND RS REGISTERS

002322  052767  000001  176636  NOWGO:  BIS     #BIT0,ONCEE     ;SET FOUND DRIVE FLAG
002330  016767  022074  176632          MOV     TIMES,TIMSV     ;SAVE TIME
002336  012767  000001  022064          MOV     #1,TIMES        ;ONLY TEST ONCE
                                ;*******************************************************
                                ;TEST 1         RESET TEST FOR REGISTERS
                                ;*******************************************************
002344  104400                  TST1:   SCOPE
002346  012737  000340  177776          MOV     #340,@#PS       ;LOCK OUT INTERUPTS
002354  016777  176600  176520          MOV     UNNUM,@RSCS2    ;LOAD UNIT #
002362  012777  177776  176510          MOV     #177776,@RSCS1  ;SET ALL
002370  012777  177777  176510          MOV     #177777,@RSBA   ;POSSIBLE R/W
002376  012777  177777  176504          MOV     #177777,@RSDA   ;BITS IN THESE REGISTERS
002404  012777  177777  176502          MOV     #177777,@RSER
002412  012777  177777  176504          MOV     #177777,@RSMR
002420  012777  177777  176456          MOV     #177777,@RSWC
002426  012777  177737  176446          MOV     #177737,@RSCS2
002434  000005                          RESET                   ;CLEAR ALL BITS IN ALL REG.

                                ;TEST RSCS2 FOR CLEARED BITS

002436  022777  000100  176436          CMP     #100,@RSCS2     ;DID THESE BITS GET CLEARED?
002444  001401                          BEQ     .+4             ;YES
002446  104200                          HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
002450  016777  176504  176424          MOV     UNNUM,@RSCS2    ;PUT # OF UNIT IN TEST IN CS2
002456  022777  010600  176426          CMP     #10600,@RSDS    ;IS DPR AND MOL SET?
002464  001401                          BEQ     .+4             ;YES
002466  104040                          HLT     !DS             ;NO WHY NOT?

                                ;TEST CONTROL AND STATUS REG 1
002470  022777  004200  176402          CMP     #4200,@RSCS1    ;DID THE READY BIT SET?
002476  001401                          BEQ     .+4             ;YES
002500  104001                          HLT     !CS1            ;READY SHOULD BE SET
```

```
1031                                    ;TEST BUS ADDRESS REGISTER
1032
1033 002502  005777  176400                     TST     @RSBA           ;IS BA REG. CLEARED
1034 002506  001401                             BEQ     .+4             ;YES
1035 002510  104020                             HLT     !BA             ;SHOULD BE 0
1036
1037                                    ;TEST DISK ADDRESS REGISTER
1038
1039 002512  005777  176372                     TST     @RSDA           ;IS DA CLEARED
1040 002516  001401                             BEQ     .+4             ;YES
1041 002520  104004                             HLT     !DA             ;SHOULD BE 0
1042
1043                                    ;TEST ERROR REG RSER
1044
1045 002522  005777  176366                     TST     @RSER           ;DID RSER CLEAR?
1046 002526  001401                             BEQ     .+4             ;YES
1047 002530  104002                             HLT     !ER             ;BITS(157015) SHOULD BE CLEARED
1048
1049                                    ;TEST RS MAINTENANCE REGISTER
1050
1051 002532  032777  000077  176364             BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
1052 002540  001401                             BEQ     .+4             ;YES
1053 002542  104220                             HLT     !MR             ;BITS(77) SHOULD BE 0
1054
1055                                    ;TEST WC REG IT SHOULD NOT CHANGE
1056
1057 002544  022777  177777  176332             CMP     #177777,@RSWC   ;DID IT CHANGE?
1058 002552  001401                             BEQ     .+4             ;NO
1059 002554  104010                             HLT     !WC             ;RESET SHOULD NOT MODIFY RSWC
1060
1061                                    ;TEST RSAS
1062
1063 002556  005777  176334                     TST     @RSAS           ;IS REG CLEAR
1064 002562  001401                             BEQ     .+4             ;YES
1065 002564  104100                             HLT     !AS             ;NO
```

```
                                         ;*****************************************************************
                                         ;TEST 2         TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
                                         ;*****************************************************************
      002566  104400                     TST2:   SCOPE

      002570  012737  000340  177776     TTAGG:  MOV     #340,@#PS               ;LOCK OUT INTERRUPTS
      002576  016777  176356  176276             MOV     UNNUM,@RSCS2
      002604  012777  043576  176266             MOV     #43576,@RSCS1           ;SET ALL
      002612  012777  177777  176266             MOV     #177777,@RSDA           ;POSSIBLE
      002620  012777  177777  176262             MOV     #177777,@RSBA           ;REGISTERS
      002626  012777  177017  176260             MOV     #177017,@RSER
      002634  012777  177777  176260             MOV     #177777,@RSDB
      002642  012777  177777  176234             MOV     #177777,@RSWC
      002650  012777  020417  176224             MOV     #20417,@RSCS2
      002656  012777  000071  176240             MOV     #71,@RSMR
      002664  012777  000040  176210             MOV     #40,@RSCS2              ;CLEAR ALL BITS
      002672  022777  000100  176202             CMP     #100,@RSCS2             ;DID THE RIGHT BITS CLEAR?
      002700  001401                             BEQ     .+4                     ;YES
      002702  104200                             HLT     !CS2                    ;(417) SHOULD BE CLEARED IN CS2
      002704  016777  176250  176170             MOV     UNNUM,@RSCS2            ;GET DRIVE NUMBER
      002712  032777  173577  176160             BIT     #173577,@RSCS1          ;DID ALL BITS GET CLEARED
      002720  001401                             BEQ     .+4                     ;YES
      002722  104001                             HLT     !CS1                    ;NO, ALL BITS SHOULD BE 0
                                         ;TEST BUS ADDRESS REGISTER

      002724  005777  176156                     TST     @RSBA                   ;IS BA REG. CLEARED
      002730  001401                             BEQ     .+4                     ;YES
      002732  104020                             HLT     !BA                     ;SHOULD BE 0

                                         ;TEST DISK ADDRESS REGISTER

      002734  005777  176150                     TST     @PSDA                   ;IS DA CLEARED
      002740  001401                             BEQ     .+4                     ;YES
      002742  104020                             HLT     !BA                     ;SHOULD BE 0

                                         ;TEST ERROR REG RSER

      002744  032777  177777  176142             BIT     #177777,@RSER           ;DID THESE BITS GET CLEARED
      002752  001401                             BEQ     .+4                     ;YES
      002754  104002                             HLT     !ER                     ;BITS(157015) SHOULD BE CLEARED

                                         ;TEST RS MAINTENANCE REGISTER
      002756  032777  000077  176140             BIT     #77,@RSMR               ;DID THESE BITS GET CLEARED
      002764  001401                             BEQ     .+4                     ;YES
      002766  104220                             HLT     !MR                     ;BITS(77) SHOULD BE 0

                                         ;TEST WC REG. IT SHOULD NOT CHANGE
      002770  022777  177777  176106             CMP     #177777,@RSWC           ;DID WC CHANGE
      002776  001401                             BEQ     .+4                     ;NO
      003000  104010                             HLT     !WC                     ;WHY DID IT CHANGE?
```

```
                                        ;*****************************************************************
                                        ;TEST 3            SET AND CLEAR ALL REGISTERS
                                        ;*****************************************************************
003002  104400                          TST3:   SCOPE
                                        ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
                                        ;BITS 7,6,5,4,3,2&1

003004  104414                                  CLRDK                           ;CLEAR ALL RS REG
003006  016767  176156  021414                  MOV     TIMSV,TIMES             ;GET TIME
003014  012777  003576  176056                  MOV     #3576,@RSCS1            ;SET DISK FUNCTION BITS
003022  022777  005776  176050                  CMP     #5776,@RSCS1            ;ARE THESE BITS SET?
003030  001401                                  BEQ     .+4                     ;NO
003032  104001                                  HLT     !CS1                    ;SHOULD = 3776
003034  012777  002524  176036                  MOV     #2524,@RSCS1            ;SET THESE BITS
003042  022777  004724  176030                  CMP     #4724,@RSCS1            ;DID THEY SET
003050  001401                                  BEQ     .+4                     ;YES
003052  104001                                  HLT     !CS1                    ;SHOULD BE 2725
003054  012777  001052  176016                  MOV     #1052,@RSCS1            ;SET THESE BITS
003062  022777  005252  176010                  CMP     #5252,@RSCS1            ;ARE THEY =?
003070  001401                                  BEQ     .+4                     ;YES
003072  104001                                  HLT     !CS1                    ;SHOULD = 1252
003074  104400                          TST4:   SCOPE
                                        ;CLEAR THE FUNCTION BITS

003076  012777  043576  175774                  MOV     #43576,@RSCS1           ;SET DISK FUNCTION BITS
003104  005077  175770                          CLR     @RSCS1
003110  022777  004200  175762                  CMP     #4200,@RSCS1            ;IS THE READY BIT SET
003116  001401                                  BEQ     .+4                     ;YES
003120  104001                                  HLT     !CS1                    ;RSCS1 SHOULD = 4200

                                        ;*****************************************************************
                                        ;TEST 5            TEST RSCS2
                                        ;*****************************************************************
003122  104400                          TST5:   SCOPE

003124  000005                                  RESET                           ;CLEAR WORLD
003126  022777  000100  175746                  CMP     #100,@RSCS2             ;DID THEY CLEAR?
003134  001401                                  BEQ     .+4                     ;YES
003136  104200                                  HLT     !CS2                    ;NO
003140  012777  021037  175734                  MOV     #21037,@RSCS2           ;SET BITS 21017
003146  022777  000137  175726                  CMP     #137,@RSCS2             ;DID THESE BITS GET SET
003154  001405                                  BEQ     1$                      ;YES
003156  017700  175720                          MOV     @RSCS2,BAD
003162  012701  000137                          MOV     #137,GOOD               ;WHAT CS2 SHOULD =
003166  104000                                  HLT                             ;CS2 = BAD GOOD = CORRECT ANS
```

```
003170  012777  020025  175704  1$:     MOV     #20025,@RSCS2           ;SET THESE BITS
003176  022777  000125  175676          CMP     #125,@RSCS2             ;DID THESE BITS GET SET
003204  001401                          BEQ     .+4                     ;YES
003206  104200                          HLT     !CS2                    ;NO CS2 SHOULD = 20125
003210  012777  000012  175664          MOV     #12,@RSCS2              ;LOAD THESE BITS
003216  022777  000112  175656          CMP     #112,@RSCS2             ;DID THESE BITS GET SET IN CS2
003224  001401                          BEQ     .+4                     ;YES
003226  104200                          HLT     !CS2                    ;BAD = CS2 GOOD = CORRECT ANS
003230  012777  177777  175644          MOV     #-1,@RSCS2              ;SET BITS
003236  005077  175640                  CLR     @RSCS2                  ;CLEAR THEM
003242  022777  000100  175632          CMP     #100,@RSCS2             ;DID CLEAR WORK
003250  001401                          BEQ     .+4                     ;YES
003252  104200                          HLT     !CS2                    ;R/W BITS DID NOT CLEAR
003254  016777  175700  175620          MOV     UNNUM,@RSCS2            ;GET UNIT #
003262  104400                  TST6:   SCOPE
                                ;CAN WE SET ALL THE RSBA BITS

003264  012777  177777  175614          MOV     #177777,@RSBA           ;SET THE BITS
003272  022777  177776  175606          CMP     #177776,@RSBA           ;DID THEY SET
003300  001401                          BEQ     .+4                     ;YES
003302  104020                          HLT     !BA                     ;BITS 17776 SHOULD BE SET
003304  012777  125252  175574          MOV     #125252,@RSBA           ;SET THESE BITS
003312  022777  125252  175566          CMP     #125252,@RSBA           ;ARE THEY =
003320  001401                          BEQ     .+4                     ;YES
003322  104020                          HLT     !BA                     ;SHOULD BE 125252
003324  012777  052524  175554          MOV     #52524,@RSBA            ;SET THESE BITS
003332  022777  052524  175546          CMP     #52524,@RSBA            ;ARE THEY =
003340  001401                          BEQ     .+4                     ;YES
003342  104020                          HLT     !BA                     ;SHOULD BE 52524

003344  104400                  TST7:   SCOPE
                                ;FLOAT A 1 THROUGH RSBA

003346  012701  000002          FLOTBA: MOV     #2,GOOD                 ;GET A 2
003352  000241                          CLC                             ;CLEAR CARRY
003354  010177  175526          1$:     MOV     GOOD,@RSBA              ;FLOAT NUMBER
003360  017700  175522                  MOV     @RSBA,BAD               ;GET BA
003364  020100                          CMP     GOOD,BAD                ;COMPARE BA
003366  001401                          BEQ     .+4                     ;BA CORRECT
003370  104000                          HLT                             ;BAD=BA GOOD=CORRECT ANS
003372  006101                          ROL     GOOD                    ;ROTATE NUMBER
003374  103367                          BCC     1$                      ;LOOP TILL DONE
```

```
003376  104400                          TST10:  SCOPE

                                        ;CLEAR THE RSBA REGISTER

003400  012777  177777  175500                  MOV     #177777,@RSBA   ;SET RSBA EQUAL TO ALL ONES
003406  005077  175474                          CLR     @RSBA
003412  005777  175470                          TST     @RSBA           ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
003416  001401                                  BEQ     .+4             ;YES
003420  104020                                  HLT     !BA             ;NO
003422  104400                          TST11:  SCOPE

                                        ;CAN WE SET ALL BITS IN RSWC REGISTER

003424  012777  177777  175452                  MCV     #177777,@RSWC   ;SET WC BITS
003432  022777  177777  175444                  CMP     #177777,@RSWC   ;ARE ALL BITS SET
003440  001401                                  BEQ     .+4             ;YES
003442  104010                                  HLT     !WC             ;NO
003444  012777  125252  175432                  MOV     #125252,@RSWC   ;SET THESE BITS
003452  022777  125252  175424                  CMP     #125252,@RSWC   ;ARE THEY =
003460  001401                                  BEQ     .+4             ;YES
003462  104010                                  HLT     !WC             ;SHOULD BE 125252
003464  012777  052525  175412                  MOV     #52525,@RSWC    ;SET THESE BITS
003472  022777  052525  175404                  CMP     #52525,@RSWC    ;ARE THEY =
003500  001401                                  BEQ     .+4             ;YES
003502  104010                                  HLT     !WC             ;SHOULD BE 152525
003504  104400                          TST12:  SCOPE

                                        ;FLOAT A 1 THROUGH RSWC

003506  012701  000001                  FLOTWC: MOV     #1,GOOD         ;GET A 1
003512  000241                                  CLC                     ;CLEAR CARRY
003514  010177  175364                  1$:     MOV     GOOD,@RSWC      ;FLOAT NUMBER
003520  017700  175360                          MOV     @RSWC,BAD       ;GET WC
003524  020100                                  CMP     GOOD,BAD        ;COMPARE WC
003526  001401                                  BEQ     .+4             ;WC CORRECT
003530  104000                                  HLT                     ;BAD=WC GOOD=CORRECT ANS
003532  006101                                  ROL     GOOD            ;ROTATE NUMBER
003534  103367                                  BCC     1$              ;LOOP TILL DONE
```

```
                                        ;CLEAR THE WORD COUNT REGISTER
 003536  104400                         TST13:  SCOPE

 003540  012777  177777  175336                 MOV     #177777,@RSWC   ;SET RSWC REGISTER EQUAL TO ALL ONES
 003546  005077  175332                         CLR     @RSWC
 003552  005777  175326                         TST     @RSWC           ;DID ALL BITS GET CLEARED
 003556  001401                                 BEQ     .+4             ;YES
 003560  104010                                 HLT     !WC             ;NO
 003562  104400                         TST14:  SCOPE

                                        ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.

 003564  012777  177777  175316                 MOV     #177777,@RSDA   ;SET ALL BITS
 003572  022777  177777  175310                 CMP     #177777,@RSDA   ;ARE THE BITS SET
 003600  001401                                 BEQ     .+4             ;YES
 003602  104004                                 HLT     !DA             ;NO
 003604  012777  125252  175276                 MOV     #125252,@RSDA   ;SET THESE BITS
 003612  022777  125252  175270                 CMP     #125252,@RSDA   ;ARE THEY =
 003620  001401                                 BEQ     .+4             ;YES
 003622  104004                                 HLT     !DA             ;SHOULD BE 125252
 003624  012777  052525  175256                 MOV     #52525,@RSDA    ;SET THESE BITS
 003632  022777  052525  175250                 CMP     #52525,@RSDA    ;ARE THEY =
 003640  001401                                 BEQ     .+4             ;YES
 003642  104004                                 HLT     !DA             ;SHOULD BE 52525
 003644  104400                         TST15:  SCOPE

                                        ;FLOAT A 1 THROUGH RSDA

 003646  012701  000001                 FLOTDA: MOV     #1,GOOD         ;GET A 1
 003652  000241                                 CLC                     ;CLEAR CARRY
 003654  010177  175230                 1$:     MOV     GOOD,@RSDA      ;FLOAT NUMBER
 003660  017700  175224                         MOV     @RSDA,BAD       ;GET DA
 003664  020100                                 CMP     GOOD,BAD        ;COMPARE DA
 003666  001401                                 BEQ     .+4             ;DA CORRECT
 003670  104000                                 HLT                     ;BAD=DA GOOD=CORRECT ANS
 003672  006101                                 ROL     GOOD            ;ROTATE NUMBER
 003674  103367                                 BCC     1$              ;LOOP TILL DONE
```

```
                                        ;CAN WE CLEAR THE RSDA REG.
 003676  104400                         TST16:  SCOPE

 003700  012777  177777  175202                 MOV     #177777,@RSDA   ;SET RSDA TO ALL ONES
 003706  005077  175176                         CLR     @RSDA
 003712  005777  175172                         TST     @RSDA           ;TEST FOR ZERO RSDA
 003716  001401                                 BEQ     .+4             ;YES
 003720  104004                                 HLT     !DA             ;ANS SHOULD BE 0
 003722  104400                         TST17:  SCOPE

                                        ;SET AND CLEAR THE RSER REG.

 003724  012777  177017  175162                 MOV     #177017,@RSER   ;SET THESE BITS
 003732  022777  177017  175154                 CMP     #177017,@RSER   ;DID THEY SET
 003740  001401                                 BEQ     .+4             ;YES
 003742  104002                                 HLT     !ER             ;RSER SHOULD = 157017
 003744  112777  000001  175142                 MOVB    #1,@RSER        ;A MOVB INST
 003752  022777  000001  175134                 CMP     #1,@RSER        ;SHOULD MODIFY COMPLETE WD
 003760  001401                                 BEQ     .+4             ;OK
 003762  104002                                 HLT     !ER

 003764  104400                         TST20:  SCOPE

 003766  012777  052005  175120                 MOV     #52005,@RSER    ;SET THESE BITS
 003774  022777  052005  175112                 CMP     #52005,@RSER    ;DID THEY SET
 004002  001401                                 BEQ     .+4             ;YES
 004004  104002                                 HLT     !ER             ;ER SHOULD = 52005
 004006  104400                         TST21:  SCOPE

 004010  012777  125012  175076                 MOV     #125012,@RSER   ;SET THESE BITS
 004016  022777  125012  175070                 CMP     #125012,@RSER   ;DID THEY SET
 004024  001401                                 BEQ     .+4             ;YES
 004026  104002                                 HLT     !ER             ;ER SHOULD = 105012
```

```
004030  104400                          TST22:  SCOPE

004032  012777  177017  175054                  MOV     #177017,@RSER   ;SET THESE BITS
004040  005077  175050                          CLR     @RSER           ;CLEAR THEM
004044  005777  175044                          TST     @RSER           ;DID THEY CLEAR
004050  001401                                  BEQ     .+4             ;YES
004052  104002                                  HLT     !ER             ;SHOULD = 0
004054  104400                          TST23:  SCOPE

                                        ;SET AND CLEAR RSMR

004056  012777  000070  175040                  MOV     #70,@RSMR       ;SET THESE BITS
004064  017767  175034  175122                  MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
004072  042767  177700  175114                  BIC     #177700,WORK    ;CLEAR JUNK
004100  022767  000070  175106                  CMP     #70,WORK        ;DID THEY SET
004106  001401                                  BEQ     .+4             ;YES
004110  104220                                  HLT     !MR             ;SHOULD = 70
004112  104400                          TST24:  SCOPE

004114  012777  000070  175002                  MOV     #70,@RSMR       ;SET BITS
004122  005077  174776                          CLR     @RSMR           ;CLEAR THEM
004126  032777  000077  174770                  BIT     #77,@RSMR       ;DID THEY CLEAR
004134  001401                                  BEQ     .+4             ;YES
004136  104220                                  HLT     !MR             ;BITS (77) SHOULD = 0
004140  104400                          TST25:  SCOPE

004142  012777  000050  174754                  MOV     #50,@RSMR       ;SET BITS
004150  017767  174750  175036                  MOV     @RSMR,WORK      ;PUT IN WORKABLE REG
004156  042767  177700  175030                  BIC     #177700,WORK    ;CLEAR JUNK
004164  022767  000050  175022                  CMP     #50,WORK        ;DID THESE BITS SET
004172  001401                                  BEQ     .+4             ;YES
004174  104220                                  HLT     !MR             ;BITS (50) SHOULD BE SET
004176  104400                          TST26:  SCOPE

004200  012777  000020  174716                  MOV     #20,@RSMR       ;SET BITS
004206  017767  174712  175000                  MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
004214  042767  177700  174772                  BIC     #177700,WORK    ;CLEAR JUNK
004222  022767  000020  174764                  CMP     #20,WORK        ;DID THEY SET
004230  001401                                  BEQ     .+4             ;YES
004232  104220                                  HLT     !MR             ;MR SHOULD AT LEAST HAVE A (21)
```

```
                                        ;*****************************************************************
                                        ;TEST 27                 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
                                        ;*****************************************************************
004234  104400                          TST27:  SCOPE

004236  104414                          BITST:  CLRDK                           ;CLEAR ALL RS REG
004240  012777  003566  174632                  MOV     #3566,@RSCS1            ;LOAD CS1
004246  112777  000005  174660                  MOVB    #5,@RSCS1B              ;LOAD BIT
004254  022777  004766  174616                  CMP     #4766,@RSCS1            ;DID IT LOAD?
004262  001401                                  BEQ     .+4                     ;YES
004264  10400!                                  HLT     !CS1
004266  112777  000032  174604                  MOVB    #32,@RSCS1
004274  022777  004632  174576                  CMP     #4632,@RSCS1
004302  001401                                  BEQ     .+4
004304  104001                                  HLT     !CS1                    ;CS1 SHOULD = 4632

004306  104400                          TST30:  SCOPE

004310  016777  174644  174564          BITCS2: MOV     UNNUM,@RSCS2            ;LOAD UNIT NUMBER
004316  052777  177400  174556                  BIS     #177400,@RSCS2          ;LOAD ALL BITS
004324  105077  174606                          CLRB    @RSCS2B                 ;CLR UPPER BYTE
004330  016701  174624                          MOV     UNNUM,GOOD              ;GET UNIT NO.
004334  052701  000100                          BIS     #100,GOOD               ;SET OR BIT
004340  017700  174536                          MOV     @RSCS2,BAD              ;GET CS2
004344  020001                                  CMP     BAD,GOOD                ;IS CS2 CORRECT?
004346  001401                                  BEQ     .+4                     ;YES
004350  104000                                  HLT                             ;LOAD BYTE DID NOT WORK

004352  104400                          TST31:  SCOPE

004354  012777  025252  174522          BITWC:  MOV     #25252,@RSWC            ;LOAD WC
004362  112777  000377  174550                  MOVB    #377,@RSWCB             ;LOAD BIT
004370  022777  177652  174506                  CMP     #177652,@RSWC           ;DID IT LOAD?
004376  001401                                  BEQ     .+4                     ;YES
004400  104010                                  HLT     !WC                     ;NO WC SHOULD =177652
004402  112777  000123  174474                  MOVB    #123,@RSWC
004410  022777  177523  174466                  CMP     #177523,@RSWC
004416  001401                                  BEQ     .+4
004420  104010                                  HLT     !WC                     ;WC SHOULD = 177523

004422  104400                          TST32:  SCOPE

004424  012777  025252  174454          BITBA:  MOV     #25252,@RSBA            ;LOAB DA
004432  112777  000377  174502                  MOVB    #377,@RSBAB             ;LOAD BIT
004440  022777  177652  174440                  CMP     #177652,@RSBA           ;DID IT LOAD?
004446  001401                                  BEQ     .+4                     ;YES
004450  104020                                  HLT     !BA                     ;DA SHOULD =177652
004452  112777  000125  174426                  MOVB    #125,@RSBA
004460  022777  177524  174420                  CMP     #177524,@RSBA
004466  001401                                  BEQ     .+4
004470  104020                                  HLT     !BA                     ;BA SHOULD = 177525
004472  104414                                  CLRDK                           ;CLEAR ALL RS REG
```

```
                                        ;*************************************************************************
                                        ;TEST 33                  LOAD RSDB WITH ALL ONES AND ALL ZEROS
                                        ;*************************************************************************
004474  104400                          TST33:  SCOPE
004476  104414                          ZERONE: CLRDK                           ;CLEAR ALL RS REG
004500  005077  174416                          CLR     @RSDB                   ;LOAD DB WITH ALL 0
004504  012777  177777  174410                  MOV     #177777,@RSDB           ;LOAD DB WITH ALL ONES
004512  012767  002000  174474                  MOV     #2000,WORK              ;TIME OUT ROUTINE
004520  012701  000300                          MOV     #300,GOOD               ;GET CORRECT FOR CS2
004524  056701  174430                          BIS     UNNUM,GOOD
004530  017700  174346                  2$:     MOV     @RSCS2,BAD              ;GET CS2
004534  020100                                  CMP     GOOD,BAD                ;IS IT CORRECT?
004536  001404                                  BEQ     3$                      ;YES
004540  005367  174450                          DEC     WORK                    ;TO WAIT FOR OR
004544  001371                                  BNE     2$                      ;TO SET
004546  104200                                  HLT     !CS2                    ;OR SHOULD BE SET
004550  005001                          3$:     CLR     GOOD
004552  017700  174344                          MOV     @RSDB,BAD               ;LOAD BAD WITH DB
004556  020100                                  CMP     GOOD,BAD                ;IS BAD CORRECT
004560  001401                                  BEQ     .+4                     ;YES
004562  104000                                  HLT                             ;COULD NOT FLOAT 0 THROUGH DB
004564  012701  177777                          MOV     #-1,GOOD                ;LOAD GOOD WITH ANS
004570  017700  174326                          MOV     @RSDB,BAD               ;GET DATA FROM DB
004574  020100                                  CMP     GOOD,BAD                ;IS DB CORRECT
004576  001401                                  BEQ     .+4                     ;YES
004600  104000                                  HLT                             ;BAD SHOULD = 1777777
```

```
                                        ;TEST INTERRUPT IN THE RH11
                                        ;BY MOVING 300 INTO RHCS1
                                        ;************************************************************************
                                        ;TEST 34                 TEST INTERRUPT IN RH11
                                        ;************************************************************************
004602  104400                          TST34:  SCOPE
004604  104414                          INT:    CLRDK                           ;CLEAR ALL ERRORS
004606  012777  004660  174314                  MOV     #PGTRAP,@RSVEC          ;SET UP VECTOR
004614  012777  000340  174310                  MOV     #340,@RSVCPS            ;SET TRAP PS
004622  012737  000200  177776                  MOV     #200,@#PS               ;SET PS AT PRIORITY 4
004630  012777  000300  174242                  MOV     #300,@RSCS1             ;THIS SHOULD CAUSE A TRAP
004636  012767  000500  174350                  MOV     #500,WORK               ;SETUP LOOP
004644  005367  174344                  1$:     DEC     WORK                    ;DEC LOOP SHOULD
004650  001375                                  BNE     1$                      ;INTERRUPT BEFORE LOOP IS DONE
004652  104001                                  HLT     !CS1                    ;SHOULD NEVER GET HERE
004654  000167  000014                          JMP     INTDON                  ;GET OUT
004660  022626                          PGTRAP: CMP     (6)+,(6)+               ;TRAP OK
004662  022777  004200  174210                  CMP     #4200,@RSCS1            ;DID IE CLEAR?
004670  001401                                  BEQ     .+4                     ;YES
004672  104001                                  HLT     !CS1                    ;IE SHOULD BE CLEARED
004674                                  INTDON:
```

```
                                        ;*************************************************************************
                                        ;TEST 35                  MAINTENANCE TIMING TEST
                                        ;*************************************************************************
 004674  104400                         TST35:  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 SUBST'TUTED 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, R&SYNC AREA, SECTOR COUNTERS, ETC.

                                        ;PUT DRIVE IN MAINTENANCE MODE
 004676  104414                         MRTIME: CLRDK                           ;CLEAR DRIVE REGISTERS
 004700  052767  001040  174260                 BIS     #1040,ONCEE             ;SET CLK CNT
 004706  104430                                 MRIND                           ;SEND INDEX PULSE TO MR REG
 004710  104420                                 MRCK                            ;CHECK MAINTENANCE REG FOR
 004712  022701                                 22701                           ;22701
 004714  104424                                 MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                                ;BY SENDING 2 CLOCK PULSES
 004716  104430                                 MRIND                           ;SEND MAINT INDEX PULSE

 004720  104420                                 MRCK                            ;CHECK MAINT REG TO
 004722  022701                                 22701                           ;EQUAL 22701
 004724  104000                                 HLT                             ;MR=BAD GOOD=CORRECTIONS
                                                                                ;COULD NOT INITIALIZE MR REG
                                        ;INDEX PULSE SHOULD CLEAR LOOK-AHEAD REG

 004726  005777  174166                         TST     @RSLA                   ;IS RSLA CLEARED
 004732  001401                                 BEQ     .+4                     ;YES
 004734  104224                                 HLT     !MR!LA                  ;RSLA SHOULD BE CLEARED
                                                                                ;WITH THE INDEX PULSE

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

 004736  012767  001000  174236                 MOV     #512.,REPT
 004744  104422                         MRTIM1: MRCLK                           ;CLOCK MAINT REG WITH AN 11 AND A 1
 004746  104420                                 MRCK                            ;CHECK MR REG TO
 004750  072701                                 72701                           ;EQUAL 72701
 004752  104000                                 HLT                             ;MR = BAD, GOOD = CORRECT ANS
 004754  104422                                 MRCLK                           ;CLOCK MR
 004756  104420                                 MRCK                            ;CHECK MR TO
 004760  022701                                 22701                           ;EQUAL 22701
 004762  104000                                 HLT                             ;BAD=MR REG GOOD=CORRECTIONS
 004764  005367  174212                         DEC     REPT                    ;IS THE LOOP DONE YET?
 004770  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

 004772  104422                                       MRCLK                        ;CLOCK MAINT REG WITH A 11 AND A 1
 004774  104420                                       MRCK                         ;CHECK MR REG TO
 004776  072301                                       72301                        ;EQUAL 72301
 005000  104000                                       HLT                          ;MR=BAD GOOD=CORRECTIONS
 005002  104422                                       MRCLK                        ;CLOCK MR WITH 11 AND A 1
 005004  104420                                       MRCK                         ;CHECK MAINT REG
 005006  022301                                       22301                        ;TO EQUAL 22301
 005010  104000                                       HLT                          ;MR=BAD GOOD-CORRECT ANS
 005012  005777  174102                               TST     @RSLA                ;DOES LOOK AHEAD REG=0
 005016  001401                                       BEQ     MRT2                 ;YES-CONT
 005020  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

 005022  005002                               MRT2:   CLR     R2                   ;CLEAR R2 FOR SECTOR COMPARE WITH LA REG
 005024  012767  000050  174150                       MOV     #40.,REPT            ;80 CLOCKS TO STEP THRU PRE-AMBLE
 005032  104422                               MRT2A:  MRCLK                        ;CLOCK MR WITH A 11 AND A 1
 005034  104420                                       MRCK                         ;CHECK MAINT REG
 005036  073701                                       73701                        ;EQUAL 73701
 005040  104000                                       HLT                          ;MR = BAD GOOD = CORRECT ANS
 005042  104422                                       MRCLK                        ;CLOCK MR REG
 005044  104420                                       MRCK                         ;CHECK MR REG
 005046  023701                                       23701                        ;TO EQUAL 23701
 005050  104000                                       HLT                          ;MR = BAD GOOD = CORRECTANS
 005052  005367  174124                               DEC REPT                     ;REPEAT
 005056  001365                                       BNE     MRT2A                ;LOOP 40 TIMES

                                              ;SUPPLY CLOCKS TO STEP THROUGH THE DATA AREA IN THE SECTOR
 005060  012767  002200  174114                       MOV     #9.*128.,REPT        ;18 CLOCKS PER DATA WORD
 005066  104422                               MRT2B:  MRCLK                        ;CLOCK MR WITH A 11 AND A 1
 005070  104420                                       MRCK                         ;CHECK MAINT REG
 005072  073701                                       73701                        ;TO EQUAL 73701
 005074  104000                                       HLT                          ;MR = BAD GOOD = CORRECT ANS
 005076  104422                                       MRCLK                        ;CLOCK MR REG
 005100  104420                                       MRCK                         ;CHECK MR REG
 005102  023701                                       23701                        ;TO EQUAL 23701
 005104  104000                                       HLT                          ;MR=BAD GOOD=CORRECTANS
 005106  005367  174070                               DEC     REPT                 ;REPEAT
 005112  001365                                       BNE     MRT2B                ;LOOP
```

```
                                             ;SUPPLY ENOUGH MAINT CLOCKS TO STEP THROUGH THE CRC AREA
                                             ;AND THE DEAD BAND ON THE SECTOR

005114  012767  000214  174060                 MOV     #140.,REPT          ;AMOUNT OF CLOCKS TO END OF SECTOR
005122  104422                         MRT2C:  MRCLK                       ;CLOCK MR WITH A 11 AND A 1
005124  104420                                 MRCK                        ;CHECK MAINT REG
005126  073701                                 73701                       ;TO EQUAL 73701
005130  104000                                 HLT                         ;MR = BAD GOOD = CORRECT ANS
005132  104422                                 MRCLK                       ;CLOCK MR REG
005134  104420                                 MRCK                        ;CHECK MAINT REG
005136  023701                                 23701                       ;TO EQUAL 23701
005140  104000                                 HLT                         ;MR=BAD GOOD=CORRECT ANS
005142  005367  174034                         DEC     REPT                ;REPEAT
005146  001365                                 BNE     MRT2C               ;LOOP
005150  104422                                 MRCLK                       ;CLOCK MR REG
005152  104420                                 MRCK                        ;CHECK MR REG
005154  073701                                 73701                       ;TO EQUAL 73701
005156  104000                                 HLT                         ;MR = BAD GOOD = CORRECT ANS
                                       ;ONE MORE CLOCK SHOULD CAUSE SECTOR PULSE
                                       ;IF NOT, CHECK E16-6

005160  104422                                 MRCLK                       ;CLOCK MR WITH A 11 AND A 1
005162  104420                                 MRCK                        ;MAINT REG SHOULD
005164  023701                                 23701                       ;EQUAL 22301
005166  104000                                 HLT                         ;MR=BAD GOOD=CORRECT ANS
005170  104422                                 MRCLK                       ;CLOCK MR WITH A 11 AND A 1
005172  104420                                 MRCK                        ;MAINT REG
005174  072301                                 72301                       ;SHOULD EQUAL 72301
005176  104000                                 HLT                         ;MR=BAD GOOD=CORRECT ANS

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

005200  022777  000002  173720                 CMP     #2,@RSDT            ;INTERLEAVED?
005206  001403                                 BEQ     3$                  ;NO
005210  062702  004000                         ADD     #4000,R2            ;YES
005214  000402                                 BR      2$                  ;CONT
005216  062702  000100                 3$:     ADD     #100,R2             ;INCREMENT SECTOR COMPARE
005222  020277  173672                 2$:     CMP     R2,@RSLA            ;LA REG SHOULD=100
005226  001401                                 BEQ     1$                  ;LA IS CORRECT
005230  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.

005232  012767  000076  173744  1$:      MOV     #62.,REPT1
005240  012767  005152  173734  MRT3:    MOV     #2666.,REPT
005246  104422                  3$:      MRCLK                      ;CLOCK MR WITH A 11 AND A 1
005250  005367  173726                   DEC     REPT               ;STEP THROUGH
005254  001374                           BNE     3$                 ;SECTOR
005256  104422                           MRCLK                      ;CLOCK MR WITH A 11 AND A 1
005260  104420                           MRCK                       ;MAINT REG
005262  022701                           22701                      ;SHOULD EQUAL 22701
005264  104000                           HLT                        ;MR=BAD GOOD=CORRECT ANS
005266  104422                           MRCLK                      ;1 MORE CLK, SAME SECTOR PULSE
005270  104420                           MRCK                       ;MAINT REG SHOULD
005272  072301                           72301                      ;EQUAL 72301
005274  104000                           HLT                        ;MR=BAD GOOD=CORRECT ANS
005276  022777  000002  173622           CMP     #2,@RSDT           ;DRIVE INTERLEAVED?
005304  001420                           BEQ     6$                 ;NO
005306  032767  001000  173652           BIT     #BIT9,ONCEE        ;DO I ADD 4000
005314  001406                           BEQ     4$                 ;OR SUBTRACT IT FROM WHAT I EXPECT TO
005316  042767  001000  173642           BIC     #BIT9,ONCEE        ;FIND IN RSLA
005324  162702  004000                   SUB     #4000,R2
005330  000406                           BR      6$
005332  052767  001000  173626  4$.      BIS     #BIT9,ONCEE
005340  062702  004000                   ADD     #4000,R2
005344  000402                           BR      5$
005346  062702  000100          6$:      ADD     #100,R2            ;INCREMENT SECTOR COMPARE
005352  017700  173542          5$:      MOV     @RSLA,BAD          ;LA REG SHOULD HAVE INCREMENTED TO NEXT SECTOR
005356  010201                           MOV     R2,GOOD            ;GET CORRECT ANS FOR RSLA
005360  020100                           CMP     GOOD,BAD           ;COMPARE FOR CORRECT ANS
005362  001401                           BEQ     1$                 ;RSLA IS GOOD
005364  104000                           HLT                        ;RSLA=BAD GOOD=CORRECT ANS

005366  005367  173612          1$:      DEC     REPT1              ;REPEAT 62
005372  001322                           BNE     MRT3               ;TIMES
005374  012767  005152  173600           MOV     #2666.,REPT        ;COUNT FOR LAST SECTOR
005402  104422                  2$:      MRCLK                      ;CLOCK
005404  005367  173572                   DEC     REPT               ;THRU
005410  001374                           BNE     2$                 ;LAST SECTOR
005412  017700  173502                   MOV     @RSLA,BAD          ;GET CONTENTS OF RSLA
005416  012701  007777                   MOV     #7777,GOOD         ;GET CORRECT ANS
005422  020100                           CMP     GOOD,BAD           ;DOES RSLA EQUAL 7777
005424  001401                           BEQ     .+4                ;YES
005426  104000                           HLT                        ;BAD=RSLA GOOD=CORRECT ANS
```

```
                                        ;*******************************************************************
                                        ;TEST 36                   SECTOR FRACTION TEST
                                        ;*******************************************************************
005430  104400                          TST36:  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.

005432  104414                          MRT4:   CLRDK                   ;CLEAR DRIVE REGISTERS
005434  052767  000040  173524                  BIS     #40,ONCEE       ;SET FLAG BITS
005442  042767  003000  173516                  BIC     #3000,ONCEE
005450  005067  173520                          CLR     MCCNT           ;CLEAR MAINT CLOCK COUNTER
005454  005002                                  CLR     R2              ;CLEAR R2 FOR SECTOR COUNTER
005456  104430                                  MRIND                   ;SEND INDEX PULSE TO MR REG
005460  104420                                  MRCK                    ;CHECK MR REG
005462  022701                                  22701                   ;TO EQUAL 22701
005464  104424                                  MRINT                   ;INIT MAINT MODE
005466  104430                                  MRIND                   ;ISSUE A MAINT INDEX PULSE
                                                                        ;TO CLEAR THE DRIVE
005470  104420                                  MRCK                    ;CHECK MAINT REG
005472  022701                                  22701                   ;TO EQUAL 22701
005474  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;ISSUE 1024 MAINT CLOCKS TO STEP THROUGH THE RESYNC AREA

005476  012767  001000  173476                  MOV     #512.,REPT      ;COUNT TO STEP THRU RESYNC AREA
005504  104422                          MRT4A:  MRCLK                   ;CLOCK THROUGH RESYNC
005506  104420                                  MRCK                    ;CHECK MAINT REG
005510  072701                                  72701                   ;TO EQUAL 72701
005512  104000                                  HLT                     ;MR = BAD GOOD = CORRECT ANS
005514  005777  173400                          TST     @RSLA           ;IS RSLA=TO 0
005520  001401                                  BEQ     .+4             ;YES
005522  104204                                  HLT     !LA             ;RSLA SHOULD=0 DURING RESPONSE
005524  104422                                  MRCLK                   ;CLOCK MR REG
005526  104420                                  MRCK                    ;CHECK MR REG
005530  022701                                  22701                   ;TO EQUAL 22701
005532  104000                                  HLT                     ;BAD=MR GOOD=CORRECT ANS
005534  005777  173360                          TST     @RSLA           ;IS RSLA=TO 0
005540  001401                                  BEQ     .+4             ;YES
005542  104204                                  HLT     !LA             ;RSLA SHOULD=0 DURING RESPONSE
005544  005367  173432                          DEC     REPT            ;LOOP THROUGH
005550  001355                                  BNE     MRT4A           ;RESYNC AREA

                                        ;ONE MORE PULSE SHOULD CAUSE THE FIRST SECTOR PULSE

005552  104422                                  MRCLK                   ;CLOCK MR WITH A 11 AND A 1
005554  104420                                  MRCK                    ;CHECK MAINT REG FOR SECTOR PULSE
005556  072301                                  72301                   ;MR SHOULD=72301
005560  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
005562  104422                          MRT4B:  MRCLK                           ;CLOCK MR REG WITH A 11 AND A 1
005564  104420                                  MRCK                            ;CHECK MAINT REG
005566  022301                                  22301                           ;TO EQUAL 22301
005570  104000                                  HLT                             ;MR=BAD GOOD=CORRECT ANS

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

005572  017700  173322                          MOV     @RSLA,BAD               ;GET RSLA
005576  010201                                  MOV     R2,GOOD                 ;GET CORRECT ANS
005600  020100                                  CMP     GOOD,BAD                ;IS THE RSLA REG CORRECT
005602  001401                                  BEQ     1$                      ;YES
005604  104000                                  HLT                             ;RSLA=BAD GOOD=COPRECTANS

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

005606  012767  000244  173366          1$:     MOV     #164.,REPT              ;FOR FIRST FRACTION CHANGE
005614  104422                          MRT4C:  MRCLK                           ;CLOCK MR REG WITH A 11 AND A 1
005616  017700  173276                          MOV     @RSLA,BAD               ;GET RSLA
005622  010201                                  MOV     R2,GOOD                 ;GET CORRECT ANS
005624  020001                                  CMP     BAD,GOOD                ;IS RSLA CORRECT
005626  001401                                  BEQ     1$                      ;YES
005630  104000                                  HLT                             ;BAD=RSLA GOOD=CORRECT ANS
005632  005367  173344                  1$:     DEC     REPT                    ;LOOP ON
005636  001366                                  BNE     MRT4C                   ;PREAMBLE AREA

                                        ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE

005640  104422                                  MRCLK                           ;CLOCK MR WITH A 11 AND A 1
005642  005202                                  INC     R2                      ;COUNT THE FRACTION
005644  017700  173250                          MOV     @RSLA,BAD               ;GET RSLA
005650  010201                                  MOV     R2,GOOD                 ;GET CORRECT ANS,
005652  020001                                  CMP     BAD,GOOD                ;IS RSLA CORRECT?
005654  001401                                  BEQ     2$                      ;YES
005656  104000                                  HLT                             ;RSLA=BAD GOOD=CORRECT ANS

                                        ;FIRST FRACTION CHANGES AFTER 164 MAINT. CLKS, THE REST
                                        ;CHANGE AFTER 40 MAINTENANCE CLOCKS

005660  012767  000076  173314          2$:     MOV     #62.,REPT               ;COUNT FOR WORDS IN A SECTOR
005666  012767  000047  173310          MRT4D:  MOV     #39.,REPT1              ;COUNT FOR SECT FRACT TO CHANGE
005674  104422                          MRT4E:  MRCLK                           ;CLOCK MR WITH A 11 AND A 1
005676  017700  173216                          MOV     @RSLA,BAD               ;GET RSLA
005702  010201                                  MOV     R2,GOOD                 ;GET CORRECT ANS
005704  020100                                  CMP     GOOD,BAD                ;IS RSLA CORRECT?
005706  001401                                  BEQ     1$                      ;YES
005710  104000                                  HLT                             ;RSLA=BAD GOOD=CORRECT ANS
005712  005367  173266                  1$:     DEC     REPT1                   ;LOOP
005716  001366                                  BNE     MRT4E
```

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

005720  104422                            MRCLK                    ;CLOCK MR WITH A 11 AND A 1
005722  022702  007777                    CMP     #7777,R2         ;AT THE LAST SECTOR-LAST FRACTION?
005726  001472                            BEQ     MRT4F            ;YES, FINISH THE SECTOR
005730  005202                            INC     R2               ;NO, ADD 1 TO FRACTION
005732  017700  173162          4$:       MOV     @RSLA,BAD        ;GET RSLA
005736  022777  000002  173162            CMP     #2,@RSDT         ;IS THIS DIRVE INTERLEAVED?
005744  001431                            BEQ     12$              ;NO
005746  032767  002000  173212            BIT     #BIT10,ONCEE     ;HAS REPT GONE TO ZERO YET FOR THIS SECTOR?
005754  001425                            BEQ     12$              ;NO
                                ;RSLA NOW POINTS TO NEXT INTERLEAVED SECTOR BIT 9 IN ONCEE
                                ;INDICATES WHETHER RSLA SHOULD NOW BE BETWEEN
                                ;0000-3700(1) OR 4000-7700(0).
005756  032767  001000  173202            BIT     #BIT9,ONCEE      ;SHOULD RSLA BE BETWEEN 0-3700?
005764  001004                            BNE     9$               ;YES
005766  052767  001000  173172            BIS     #BIT9,ONCEE      ;SET FOR NEXT PASS
005774  000406                            BR      10$
005776  042767  001000  173162  9$:       BIC     #BIT9,ONCEE      ;CLEAR FOR NEXT PASS
006004  042702  004000                    BIC     #4000,R2         ;MAKE EXPECTED RSLA LESS THAN 4000
006010  000404                            BR      5$
006012  062702  004000          10$:      ADD     #4000,R2         ;COMPENSATE FOR INTERLEAVING
006016  162702  000100                    SUB     #100,R2
006022  042767  002000  173136  5$:       BIC     #BIT10,ONCEE     ;CLEAR FLAG FOR NEXT SECTOR
006030  010201                  12$:      MOV     R2,GOOD          ;GET CORRECT ANSWER FOR RSLA
006032  020100                            CMP     GOOD,BAD         ;IS RSLA CORRECT
006034  001401                            BEQ     2$               ;YES
006036  104000                            HLT                      ;RSLA=BAD GOOD=CORRECT ANS
006040  005367  173136          2$:       DEC     REPT             ;HAS SECTOR FRACTION REACHED 77?
006044  001310                            BNE     MRT4D            ;NO

                                ;CHECK FOR END OF ONE SECTOR OR BEGINNING OF NEXT

006046  010203                  11$:      MOV     R2,R3
006050  042703  177700                    BIC     #177700,R3       ;CHECK SECTOR FRACTION
006054  022703  000077                    CMP     #77,R3           ;END OF SECTOR?
006060  001402                            BEQ     3$               ;YES
006062  000167  177474                    JMP     MRT4B            ;NO, BEGINNING OF NEXT
006066  012767  000025  173110  3$:       MOV     #21.,REPT1       ;SETUP LOOP TO FINISH
006074  012767  000001  173100            MOV     #1,REPT          ;THIS SECTOR
006102  052767  002000  173056            BIS     #BIT10,ONCEE     ;REPT HAS GONE TO ZERO FOR THIS SECTOR
006110  000167  177560                    JMP     MRT4E            ;LOOP

006114  012767  000021  173060  MRT4F:    MOV     #17.,REPT
006122  104422                  1$:       MRCLK                    ;CLOCK MR WITH A 11 AND A 1
006124  017700  172770                    MOV     @RSLA,BAD        ;GET RSLA
006130  010201                            MOV     R2,GOOD          ;R2 SHOULD=7777
006132  020100                            CMP     GOOD,BAD         ;IS RSLA CORRECT-END OF DISK?
006134  001401                            BEQ     2$               ;YES
006136  104000                            HLT                      ;RSLA=BAD GOOD=CORRECT ANS (7777)
006140  005367  173036          2$:       DEC     REPT             ;FINISH
006144  001366                            BNE     1$               ;LOOP
```

```
                                 ;SECTOR AND FRACTION IS = TO 7777 TO INDICATE LAST WORD ON THIS TRACK
                                 ;RSLA SHOULD EQUAL 7700 ON ANOTHER MAINT CLOCK.

006146  104422                   MRT4G:  MRCLK                   ;CLOCK MR WITH A 11 AND A 1
006150  017700  172744                   MOV     @RSLA,BAD       ;GET RSLA
006154  012701  007700                   MOV     #7700,GOOD      ;GET CORRECT ANS
006160  020100                           CMP     GOOD,BAD        ;IS RSLA CORRECT?
006162  001401                           BEQ     1$              ;YES
006164  104000                           HLT                     ;RSLA=BAD GOOD=CORRECT ANS
006166  104430                   1$:     MRIND                   ;ISSUE AN INDEX PULSE TO
                                                                 ;CLEAR THE DRIVE
006170  017700  172724                   MOV     @RSLA,BAD       ;GET RSLA
006174  005001                           CLR     GOOD            ;GET CORRECT ANS
006176  020100                           CMP     GOOD,BAD        ;IS RSLA CORRECT?
006200  001401                           BEQ     2$              ;YES
006202  104000                           HLT                     ;RSLA=BAD GOOD=CORRECT ANS
006204  104420                   2$:     MRCK                    ;CHECK MR REG
006206  022701                           22701                   ;TO EQUAL 22701
006210  104000                           HLT                     ;MR=BAD GOOD=CORRECT ANS
```

```
                                ;*******************************************************************************
                                ;TEST 37                    ILLEGAL FUNCTION TEST
                                ;*******************************************************************************
006212  104400                  TST37:  SCOPE

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

006214  104414                  MRILF:  CLROK                     ;CLEAR ALL THE DRIVE REGISTERS
006216  042767  000040  172742          BIC     #BIT5,ONCEE       ;CLEAR CLOCK CNT FLAG
006224  032767  000002  172734          BIT     #BIT1,ONCEE       ;WAS THERE AN ERROR
006232  001002                          BNE     MRLF1             ;YES DO NCT CHANGE "I'LF" CODE
006234  012702  000003                  MOV     #3,R2             ;SETUP FIRST "ILF" CODE
                                ;PUT DRIVE IN MAINTENANCE MODE

006240  104416                  MRLF1:  MROMD                     ;PUT DRIVE INTO MAINT MODE
006242  104420                          MRCK                      ;CHECK MR REG TO
006244  022701                          22701                     ;EQUAL 22701
006246  104424                          MRINT                     ;INIT MAINT MODE (CLEAR MRSP)

                                ;ASSERT A MAINTENANCE MODE DISK "INDEX" PULSE

006250  104430                          MRIND
006252  010277  172622          MRLF2:  MOV     R2,@RSCS1         ;SEND "ILF" WITH THE "GO" BIT
006256  017700  172630                  MOV     @RSDS,BAD         ;GET DRIVE STATUS REG
006262  012701  150600                  MOV     #150600,GOOD      ;GET CORRECT ANS
006266  020100                          CMP     GOOD,BAD          ;IS RSDS CORRECT?
006270  001440                          BEQ     1$                ;YES
006272  104402  006276                  TYPE    ,.+2              ;.ASCIZ <15><12>"ILLEGAL FUNCTION CODE SENT TO DRIVE= "
006346  010267  172642                  MOV     R2,WORK           ;GET FUNCTION CODE
006352  016746  172636                  MOV     WORK,-(6)         ;PUT WORK ON STACK
006356  104406                          TYPES                     ;TYPE STACK IN OCTAL - SUPRESS
006360  052767  000002  172600          BIS     #BIT1,ONCEE       ;SET ERROR BIT SO ILLEGAL FUN DOESN'T CHANGE
006366  104000                          HLT                       ;RSDS=BAD GOOD=CORRECT ANS
006370  104040                          HLT     !DS

006372  042767  000002  172566  1$:     BIC     #BIT1,ONCEE       ;CLEAR ERROR FLAG
006400  017700  172510                  MOV     @RSER,BAD         ;GET RSER
006404  012701  000001                  MOV     #1,GOOD           ;GET CORRECT ANS
006410  020100                          CMP     GOOD,BAD          ;DID "ILF" SET IN RSER
006412  001404                          BEQ     2$                ;YES
006414  052767  000002  172544          BIS     #BIT1,ONCEE       ;SET ERROR BIT
006422  104000                          HLT                       ;RSER=BAD GOOD=CORRECT ANS
006424  042767  000002  172534  2$:     BIC     #BIT1,ONCEE       ;CLEAR ERROR FLAG
```

```
                                        ;CLEAR THE DRIVE FOR THE NEXT "ILF" CODE PASS
006432  104414                          MRCILF: CLRDK                   ;CLEAR ERRORS
006434  017700  172452                          MOV     @RSDS,BAD       ;GET RSDS REG
006440  012701  010600                          MOV     #10600,GOOD     ;GET CORRECT ANS
006444  020100                                  CMP     GOOD,BAD        ;DID "ATA" AND "ERR" CLEAR IN RSDS?
006446  001435                                  BEQ     1$              ;YES
006450  104402  006454                          TYPE    ,.+2            ;.ASCIZ <15><12>"ATA AND ERR IN RSDS SHOULD CLEAR WITH I
006532  052767  000002  172426                  BIS     #BIT1,ONCEE
006540  104000                                  HLT                     ;RSDS=BAD GOOD=CORRECT ANS
006542  042767  000002  172416  1$:             BIC     #BIT1,ONCEE     ;CLEAR ERROR FLAG
006550  017700  172340                          MOV     @RSER,BAD       ;GET RSER
006554  005001                                  CLR     GOOD            ;GET CORRECT ANS
006556  020100                                  CMP     GOOD,BAD        ;DID ILF CLEAR IN RSER
006560  001431                                  BEQ     2$              ;YES
006562  052767  000002  172376                  BIS     #BIT1,ONCEE     ;SET ERROR BIT
006570  104402  006574                          TYPE    ,.+2            ;.ASCIZ <15><12>"ILF IN RSER SHOULD CLEAR WITH INIT"
006642  104000                                  HLT                     ;RSER=BAD GOOD=CORRECT ANS
006644  042767  000002  172314  2$:             BIC     #BIT1,ONCEE     ;CLEAR ERROR BIT
                                        ;GET NEXT ILLEGAL FUNCTION COE

006652  062702  000002                  MRLF3:  ADD     #2,R2           ;UPDATE ILF
006656  022702  000011                          CMP     #11,R2          ;IS THIS A ILF CODE
006662  001773                                  BEQ     MRLF3           ;NO-UPDATE IT
006664  022702  000021                          CMP     #21,R2
006670  001770                                  BEQ     MRLF3
006672  022702  000031                          CMP     #31,R2
006676  001765                                  BEQ     MRLF3
006700  022702  000051                          CMP     #51,R2
006704  001762                                  BEQ     MRLF3
006706  022702  000061                          CMP     #61,R2
006712  001757                                  BEQ     MRLF3
006714  022702  000071                          CMP     #71,R2
006720  001754                                  BEQ     MRLF3
006722  022702  000101                          CMP     #101,R2
006726  001402                                  BEQ     ILFDON          ;FINISHED ALL ILF CODES GET OUT
006730  000167  177304                          JMP     MRLF1           ;START NEXT ILF FUNCTION
006734                                  ILFDON:
```

```
                                        ;***********************************************************************
                                        ;TEST 40                TEST NO-OP CODES 1 AND 21
                                        ;***********************************************************************
 006734  104400                         TST40:  SCOPE

                                        ;MODULE TESTED M7759
 006736  104414                         MROP:   CLRDK                           ;CLEAR ALL DRIVE REGISTERS
 006740  042767  000004  172220                 BIC     #BIT2,ONCEE             ;CLEAR ERROR FLAG
 006746  104416                                 MROMD                           ;PUT DRIVE INTO MAINT MODE
 006750  104420                                 MRCK                            ;CHECK MR REG TO
 006752  022701                                 22701                           ;EQUAL 22701
 006754  104424                                 MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                                ;SEND INDEX PULSE
 006756  032767  000010  172202                 BIT     #BIT3,ONCEE             ;TESTING CODE I
 006764  001031                                 BNE     3$                      ;NO CODE 21
 006766  012777  000001  172104                 MOV     #1,@RSCS1               ;LOAD NO-OP FUNCTION
 006774  012767  000001  172212                 MOV     #1,WORK                 ;LOAD NO-OP FUNCTION
 007002  005777  172106                         TST     @RSER                   ;ANY ERRORS
 007006  001403                                 BEQ     1$                      ;NO
 007010  004767  012254                         JSR     PC,NOPERR               ;TYPE IT
 007014  104040                                 HLT     !DS                     ;TYPE ERROR
 007016  022777  010600  172066 1$:             CMP     #10600,@RSDS            ;IS RSDS CORRECT
 007024  001403                                 BEQ     2$                      ;YES
 007026  004767  012236                         JSR     PC,NOPERR               ;RSDS SHOULD
 007032  104040                                 HLT     !DS                     ;EQUAL 10600
 007034  042767  000004  172124 2$:             BIC     #BIT2,ONCEE             ;CLEAR ERROR FLAG

                                        ;TEST NO-OP FUNCTION CODE 21

 007042  052767  000010  172116                 BIS     #BIT3,ONCEE             ;TEST TESTING CODE 21 FLAG
 007050  012767  000021  172136 3$:             MOV     #21,WORK                ;LOAD CODE 21
 007056  012777  000021  172014                 MOV     #21,@RSCS1              ;LOAD FUNCTION
 007064  005777  172024                         TST     @RSER                   ;ANY ERRORS?
 007070  001403                                 BEQ     4$                      ;NO
 007072  004767  012172                         JSR     PC,NOPERR               ;YES, TYPE ERROR
 007076  104040                                 HLT     !DS                     ;ERROR DURING NO-OP FUNCTION
 007100  022777  010600  172004 4$:             CMP     #10600,@RSDS            ;IS RSDS CORRECT
 007106  001403                                 BEQ     5$                      ;YES
 007110  004767  012154                         JSR     PC,NOPERR               ;TYPE ERROR
 007114  104040                                 HLT     !DS                     ;RSDS SHOULD=10600
 007116  042767  000014  172042 5$:             BIC     #14,ONCEE               ;CLEAR TEST BITS
```

```
                                        ;************************************************************************
                                        ;TEST 41                  TEST NO-OP FUNCTION WITH ERROR BITS SET
                                        ;************************************************************************
007124  104400                          TST41:  SCOPE

                                        ;MODULE TESTED M7759
007126  104414                          MROPER: CLROK                         ;CLEAR ALL REGISTERS
007130  104416                                  MRDMD                         ;PUT DRIVE INTO MAINT MODE
007132  104420                                  MRCK                          ;CHECK MR REG
007134  022701                                  22701                         ;TO EQUAL 22701
007136  104424                                  MRINT                         ;INIT MAINT MODE (CLEAR MRSP)
007140  104430                                  MRIND                         ;SEND INDEX PULSE

007142  012777  177777  171744                  MOV     #-1,@RSER             ;LOAD RSER WITH ERRORS
007150  116701  172010                          MOVB    UNCMP,GOOD            ;GET DRIVE UNDER TEST
007154  017700  171736                          MOV     @RSAS,BAD             ;GET RSAS REG
007160  020100                                  CMP     GOOD,BAD              ;DID ATA BIT SET CAUSED BY ERROR
007162  001427                                  BEQ     1$                    ;YES
007164  104402  007170                          TYPE    ,.+2                  ;.ASCIZ <15><12>"SET ERRORS IN RSER-RSAS IS INCORRECT"
007240  104000                                  HLT                           ;RSAS=BAD GOOD=CORRECT ANS
007242  012767  000001  171744  1$:     MOV     #1,WORK               ;SETUP FOR NO-OP CODE 1
007250  032767  000010  171710          BIT     #BIT3,ONCEE           ;TESTING CODE 21?
007256  001004                                  BNE     2$                    ;YES
007260  012777  000001  171612          MOV     #1,@RSCS1             ;SEND NO-OP CODE 1
007266  000406                                  BR      3$                    ;CHECK FOR ERRORS
007270  012767  000021  171716  2$:     MOV     #21,WORK              ;SETUP FOR CODE 21
007276  012777  000021  171574          MOV     #21,@RSCS1            ;SENT NO-OP CODE 21
007304  017700  171604          3$:     MOV     @RSER,BAD             ;GET RSER REG
007310  012701  177017                          MOV     #177017,GOOD          ;GET CORRECT ANS
007314  020100                                  CMP     GOOD,BAD              ;DID RSER CHANGE WITH NO-OP
007316  001411                                  BEQ     4$                    ;NO
007320  104402  007324                          TYPE    ,.+2                  ;.ASCIZ <15><12>"RSER "
007334  004767  012024                          JSR     PC,CHG
007340  104000                                  HLT                           ;RSER=BAD GOOD=CORRECT ANS
007342  017700  171550          4$:     MOV     @RSAS,BAD             ;GET RSAS
007346  116701  171612                          MOVB    UNCMP,GOOD            ;GET CORRECT ANS
007352  020100                                  CMP     GOOD,BAD              ;IS RSAS CORRECT
007354  001411                                  BEQ     5$                    ;YES
007356  104402  007362                          TYPE    ,.+2                  ;.ASCIZ <15><12>"RSAS "
007372  004767  011766                          JSR     PC,CHG                ;TYPE ERROR
007376  104000                                  HLT                           ;RSAS=BAD GOOD=CORRECT ANS
007400  017700  171506          5$:     MOV     @RSDS,BAD             ;GET RSDS
007404  012701  150600                          MOV     #150600,GOOD          ;GET CORRECT ANS
007410  020100                                  CMP     GOOD,BAD              ;DID RSDS CHANGE
007412  001411                                  BEQ     6$                    ;NO
007414  104402  007420                          TYPE    ,.+2                  ;.ASCIZ <15><12>"RSDS "
007430  004767  011730                          JSR     PC,CHG                ;TYPE ERROR
007434  104000                                  HLT                           ;RSDS=BAD GOOD=CORRECT ANS
007436  032767  000010  171522  6$:     BIT     #BIT3,ONCEE           ;TESTING CODE 21
007444  001005                                  BNE     7$                    ;YES, GET OUT
007446  052767  000010  171512          BIS     #BIT3,ONCEE           ;SET CODE 21 FLAG
007454  000167  177446                          JMP     MROPER                ;TEST CODE 21
007460  042767  000010  171500  7$:     BIC     #BIT3,ONCEE           ;DONE CLEAR FLAG AND CONT.
```

```
                                ;********************************************************************
                                ;TEST 42                 BLOCK SEARCH TEST 1
                                ;********************************************************************
007466  104400                  TST42:  SCOPE

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

007470  104414                  MRSRCH: CLRDK                   ;CLEAR ALL REGISTERS
007472  052767  000040  171466          BIS     #BIT5,ONCEE     ;SET CLOCK FLAG
007500  104416                          MROMD                   ;PUT DRIVE INTO MAINTENANCE MOE
007502  104420                          MRCK                    ;CHECK MR REG
007504  022701                          22701                   ;TO EQUAL 22701
007506  104424                          MRINT                   ;INIT MR REG (CLEAR MRSP)
007510  104430                          MRIND                   ;CLOCK INDEX PULSE IN RSMR
007512  012777  000003  171370          MOV     #3,@RSDA        ;DO A SEARCH FOR SECTOR 3 OR 41
007520  022777  000002  171400          CMP     #2,@RSDT        ;INTERLEAVED?
007526  001403                          BEQ     4$              ;NO SECTOR 3
007530  012777  000041  171352          MOV     #41,@RSDA       ;YES SECTOR 41
007536  012777  000031  171334  4$:     MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759)
007544  104426                          DSCK                    ;CHECK RSDS
007546  030400                          30400                   ;TO EQUAL 30400
007550  104000                          HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                                ;BE 0 FOR A DRIVE SEARCH CMD
007552  012767  021506  171422          MOV     #21506,REPT     ;STEP THROUGH 3 SECTORS
007560  104422                  1$:     MRCLK                   ;CLOCK MR
007562  104426                          DSCK                    ;RSDS SHOULD NOT
007564  030400                          30400                   ;CHANGE TILL CLOCKING IS COMPLETED
007566  104000                          HLT                     ;TO REACH SECTOR 3
007570  005367  171406                  DEC     REPT            ;KEEP CLOCKING TILL
007574  001371                          BNE     1$              ;SECTOR 3 HAS BEEN REACHED
                                ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
007576  104422                          MRCLK                   ;CLOCK MR REG
007600  104426                          DSCK                    ;CHECK FOR "ATA" AND "DRY"
007602  110600                          110600                  ;TO BE SET IN RSDS FOR
007604  104000                          HLT                     ;SEARCH FUNCTION SHOULD BE COMPLETED
007606  022777  104230  171264          CMP     #104230,@RSCS1  ;SET RSCS1
007614  001401                          BEQ     2$              ;SC IN RSCS1 SHOULD SET BECAUSE OF
007616  104140                          HLT     !DS!AS          ;COMPLETED SEARCH FUNCTION
007620  016777  171336  171270  2$:     MOV     UNITSV,@RSAS    ;CLEAR ATA
007626  005777  171264                  TST     @RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
007632  001401                          BEQ     3$              ;YES
007634  104140                          HLT     !DS!AS          ;RSAS SHOULD=0
007636  022777  004230  171234  3$:     CMP     #4230,@RSCS1    ;DID SC CLEAR BY CLEARING
007644  001401                          BEQ     .+4             ;"ATA" YES
007646  104140                          HLT     !DS!AS          ;NO
```

```
                                ;*************************************************************
                                ;TEST 43                 BLOCK SEARCH TEST 2
                                ;*************************************************************
007650  104400                  TST43:  SCOPE

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

007652  104414                  MRSRC:  CLROK                   ;CLEAR ALL REGISTERS
007654  052767  000040  171304          BIS     #BITS,ONCEE     ;SET CLOCK FLAG
007662  104416                          MROMD                   ;PUT DRIVE INTO MAINTENANCE MOE
007664  104420                          MRCK                    ;CHECK MR REG
007666  022701                          22701                   ;TO EQUAL 22701
007670  104424                          MRINT                   ;INIT MR REG (CLEAR MRSP)
                                ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
007672  104430                          MRIND
007674  104420                          MRCK                    ;CHECK MR REG TO EQUAL
007676  022701                          22701                   ;22701
007700  104000                          HLT                     ;
                                ;STEP THRU RESYNC PERIOD
007702  012767  001000  171272          MOV     #512.,REPT
007710  052767  000040  171250          BIS     #BITS,ONCEE     ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
007716  104422                  MRRT1:  MRCLK                   ;CLOCK MR REG
007720  104420                          MRCK                    ;CHECK FOR
007722  072701                          72701                   ;CORRECT DATA
007724  104000                          HLT                     ;MR = BAD GOOD = CORRECT DATA
007726  104422                          MRCLK                   ;CLOCK MR REG
007730  104420                          MRCK                    ;CHECK FOR
007732  022701                          22701                   ;CORRECT DATA
007734  104000                          HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
007736  005367  171240                  DEC     REPT            ;FINISH LOOPING
007742  001365                          BNE     MRRT1           ;THROUGH RESYNC PERIOD
                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE SP = 0
007744  104422                          MRCLK                   ;CLOCK MR REG
007746  104420                          MRCK                    ;MR SHOULD
007750  072301                          72301                   ;EQUALS 72301
007752  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
007754  104422                          MRCLK                   ;CLOCK MR REG
007756  104420                          MRCK                    ;CHECK MR
007760  022301                          22301                   ;TO EQUAL 22301
007762  104000                          HLT                     ;MR=BAD GOOD=CORRECT ANS
007764  012767  000100  171210          MOV     #100,REPT       ;STEP INTO SECTOR 0
007772  104422                  2$:     MRCLK                   ;CLOCK MR REG
007774  005367  171202                  DEC     REPT            ;DO 100 TIMES
010000  001374                          BNE     2$              ;DONE YET? NO BR
010002  012777  000031  171070  4$:     MOV     #31,@RSCS1      ;LOAD SEARCH COMMAND (M7759) FOR SECTOR 0
010010  104426                          DSCK                    ;CHECK RSDS
010012  030400                          30400                   ;TO EQUAL 30400
010014  104000                          HLT                     ;PIP SHOULD BE SET AND DRY SHOULD
                                                                ;BE 0 FOR A DRIVE SEARCH CMD
010016  012767  021506  171156          MOV     #21506,REPT     ;STEP 3 SECTORS BEYOND SECTOR 0
```

```
010024  104422                           1$:     MRCLK                           ;CLOCK MR
010026  104426                                   DSCK                            ;RSDS SHOULD NOT
010030  030400                                   30400                           ;CHANGE TILL CLOCKING IS COMPLETED
010032  104000                                   HLT                             ;TO REACH SECTOR 3
010034  005367  171142                           DEC     REPT                    ;KEEP CLOCKING TILL
010040  001371                                   BNE     1$                      ;SECTOR 3 HAS BEEN REACHED
                                         ;ASSERT INDEX PULSE TO SIMULATE THE BEGINNING OF THE NEXT REVOLUTION
010042  104430                                   MRIND
010044  104420                                   MRCK                            ;CHECK MR REG TO EQUAL
010046  022701                                   22701                           ;22701
010050  104000                                   HLT

                                         ;STEP THRU RESYNC PERIOD

010052  012767  001000  171122                   MOV     #512.,REPT
010060  052767  000040  171100                   BIS     #BIT5,ONCEE             ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
010066  104422                           MRWR1:  MRCLK                           ;CLOCK MR REG
010070  104420                                   MRCK                            ;CHECK FOR
010072  072701                                   72701                           ;CORRECT DATA
010074  104000                                   HLT                             ;MR = BAD GOOD = CORRECT DATA
010076  104422                                   MRCLK                           ;CLOCK MR REG
010100  104420                                   MRCK                            ;CHECK FOR
010102  022701                                   22701                           ;CORRECT DATA
010104  104000                                   HLT                             ;ERROR WHILE CLOCKING  THROUGH RESYNC PERIOD
010106  005367  171070                           DEC     REPT                    ;FINISH LOOPING
010112  001365                                   BNE     MRWR1                   ;THROUGH RESYNC PERIOD

                                         ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                         ;SP=0 EQUALS SECTOR PULSE
010114  104422                                   MRCLK                           ;CLOCK MR REG
010116  104420                                   MRCK                            ;MR SHOULD
010120  072301                                   72301                           ;EQUAL 72301
010122  104000                                   HLT                             ;MR=BAD GOOD=CORRECT ANS
010124  104422                                   MRCLK                           ;CLOCK MR REG
010126  104420                                   MRCK                            ;CHECK MR
010130  022301                                   22301                           ;TO EQUAL 22301
010132  104000                                   HLT                             ;MR=BAD GOOD=CORRECT ANS

                                         ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
010134  104422                                   MRCLK                           ;CLOCK MR REG
010136  104426                                   DSCK                            ;CHECK FOR "ATA" AND "DRY"
010140  110600                                   110600                          ;TO BE SET IN RSDS FOR
010142  104000                                   HLT                             ;SEARCH FUNCTION SHOULD BE COMPLETED
010144  022777  104230  170726                   CMP     #104230,@RSCS1          ;SET RSCS1
010152  001401                                   BEQ     2$                      ;SC IN RSCS1 SHOULD SET BECAUSE OF
010154  104140                                   HLT     !DS!AS                  ;COMPLETED SEARCH FUNCTION
010156  016777  171000  170732  2$:              MOV     UNITSV,@RSAS            ;CLEAR ATA
010164  005777  170726                           TST     @RSAS                   ;DID ATA CLEAR BY WRITING INTO IT?
010170  001401                                   BEQ     3$                      ;YES
010172  104140                                   HLT     !DS!AS                  ;RSAS SHOULD=0
010174  022777  004230  170676  3$:              CMP     #4230,@RSCS1            ;DID SC CLEAR BY CLEARING
010202  001401                                   BEQ     .+4                     ;"ATA" YES
010204  104140                                   HLT     !DS!AS  ;NO
```

```
                                        ;*******************************************************************
                                        ;TEST 44                  DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)
                                        ;*******************************************************************
 010206  104400                         TST44:  SCOPE

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

 010210  104414                         RMRC1:  CLRDK                           ;CLEAR ALL DRIVE REGISTERS
 010212  042767  000040  170746                 BIC     #BITS,ONCEE             ;CLEAR CLK CNT FLAG
 010220  104416                                 MROMD                           ;PUT DRIVE INTO MAINT MODE
 010222  104420                                 MRCK                            ;CHECK MR REG TO
 010224  022701                                 22701                           ;EQUAL 22701
 010226  104424                                 MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
 010230  012777  000001  170652                 MOV     #1,@RSDA                ;LOAD RSDA
 010236  012777  000031  170634                 MOV     #31,@RSCS1              ;LOAD BLOCK SEARCH FUNCTION
 010244  104426                                 DSCK                            ;CHECK RSDS
 010246  030400                                 30400                           ;TO EQUAL 30400
 010250  104000                                 HLT                             ;DRY IN RSDS SHOULD BE
                                                                                ;CLEAR FOR DRIVE WAS
                                                                                ;ISSUED A BLOCK SEARCH FUNCTION
                                                                                ;RSDS=BAD GOOD=CORRECT ANS
 010252  012777  000011  170620                 MOV     #11,@RSCS1              ;LOAD A CLEAR FUNCTION
                                                                                ;THIS SHOULD CAUSE AN RMR
                                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                                ;WHEN CLEAR COMMAND WAS GIVEN
 010260  017700  170630                         MOV     @RSER,BAD               ;GET RSER REG
 010264  012701  000004                         MOV     #4,GOOD                 ;GET CORRECT ANS
 010270  020100                                 CMP     GOOD,BAD                ;DID RMR SET IN RSER?
 010272  001410                                 BEQ     1$                      ;YES
 010274  104402  021435                         TYPE    ,TRMR
 010300  104402  010304                         TYPE    ,.+2                    ;.ASCIZ "RSCS1"
 010312  104000                                 HLT                             ;RSER=BAD GOOD=CORRECT ANS
 010314  104426                         1$:     DSCK                            ;CHECK RSDS TO
 010316  150600                                 150600                          ;EQUAL 150600
 010320  104000                                 HLT                             ;RSDS=BAD GOOD=CORRECT ANS
 010322  022777  104230  170550                 CMP     #104230,@RSCS1          ;DID CORRECT BITS SET IN RSCS1
 010330  001401                                 BEQ     2$                      ;YES
 010332  104040                                 HLT     !DS                     ;RSCS1 SHOULD=104230
                                                                                ;RSDS SHOULD=150600
                                                                                ;RSER SHOULD=4
 010334  022777  000001  170546         2$:     CMP     #1,@RSDA                ;DID CLR CLEAR RSDA
 010342  001401                                 BEQ     4$                      ;NO
 010344  104004                                 HLT     !DA                     ;RSDA SHOULD=1
 010346  104414                         4$:     CLRDK                           ;CLEAR ALL REGISTERS
 010350  005777  170540                         TST     @RSER                   ;RSER SHOULD CLEAR
 010354  001401                                 BEQ     3$                      ;RSER OK
 010356  104040                                 HLT     !DS                     ;RSER SHOULD=0 FOR THE
                                                                                ;CLEAR BIT WAS LOADED IN RSCS2
 010360  022777  004200  170512         3$:     CMP     #4200,@RSCS1            ;RSCS1 SHOULD=4200 FOR THE
 010366  001401                                 BEQ     .+4                     ;CLEAR BIT WAS LOADED IN RSCS2
 010370  104040                                 HLT     !DS                     ;RSCS1 SHOULD=4200
```

```
                                ;*************************************************************
                                ;TEST 45                 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)
                                ;*************************************************************
010372  104400                  TST45:  SCOPE

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

010374  104414                  RMRC2:  CLROK                   ;CLEAR ALL DRIVE REGISTERS
010376  104416                          MRDMD                   ;PUT DRIVE INTO MAINT MODE
010400  104420                          MRCK                    ;CHECK MR REG TO
010402  022701                          22701                   ;EQUAL 22701
010404  104424                          MRINT                   ;INIT MAINT MODE (CLEAR MRSP)
010406  012777  000001  170474          MOV     #1,@RSDA        ;LOAD RSDA
010414  012777  000031  170456          MOV     #31,@RSCS1      ;LOAD BLOCK SEARCH FUNCTION
010422  104426                          DSCK                    ;CHECK RSDS
010424  030400                          30400                   ;TO EQUAL 30400
010426  104000                          HLT                     ;DRY IN RSDS SHOULD BE
                                                                ;CLEARED FOR DRIVE WAS
                                                                ;ISSURED A BLOCK SEARCH FUNCTION
                                                                ;RSDS=BAD GOOD=CORRECT ANS
010430  005077  170454                  CLR     @RSDA           ;MODIFY RSDA
                                                                ;THIS SHOULD CAUSE AN RMR
                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                ;WHEN COMMAND WAS GIVEN
010434  017700  170454                  MOV     @RSER,BAD       ;GET RSER REG
010440  012701  000004                  MOV     #4,GOOD         ;GET CORRECT ANS
010444  020100                          CMP     GOOD,BAD        ;DID RMR SET IN RSER?
010446  001410                          BEQ     1$              ;YES
010450  104402  021435                  TYPE    ,TRMR
010454  104402  010460                  TYPE    ,.+2            ;.ASCIZ "RSDA"
010466  104000                          HLT                     ;RSER=BAD GOOD=CORRECT ANS
010470  104426                  1$:     DSCK                    ;CHECK RSDS TO
010472  150600                          150600                  ;EQUAL 150600
010474  104000                          HLT                     ;RSDS=BAD GOOD=CORRECT ANS
010476  022777  104230  170374          CMP     #104230,@RSCS1  ;DID CORRECT BITS SET IN RSCS1
010504  001401                          BEQ     2$              ;YES
010506  104040                          HLT     !DS             ;RSCS1 SHOULD=104230
                                                                ;RSDS SHOULD=50400
                                                                ;RSER SHOULD=4
010510  022777  000001  170372  2$:     CMP     #1,@RSDA        ;DID CLR CLEAR RSDA
010516  001401                          BEQ     4$              ;NO
010520  104004                          HLT     !DA             ;RSDA SHOULD=1
010522  104414                  4$:     CLROK                   ;CLEAR ALL REGISTERS
010524  005777  170364                  TST     @RSER           ;RSER SHOULD CLEAR
010530  001401                          BEQ     3$              ;RSER OK
010532  104040                          HLT     !DS             ;RSER SHOULD=0 FOR THE
                                                                ;CLEAR BIT WAS LOADED IN RSCS2
010534  022777  004200  170336  3$:     CMP     #4200,@RSCS1    ;RSCS1 SHOULD=4200 FOR THE
010542  001401                          BEQ     .+4             ;CLEAR BIT WAS LOADED IN RSCS2
010544  104040                          HLT     !DS             ;RSCS1 SHOULD=4200
```

```
                                        ;*********************************************************************
                                        ;TEST 46               DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)
                                        ;*********************************************************************
010546  104400                          TST46:  SCOPE

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

010550  104414                          RMRC3:  CLROK                           ;CLEAR ALL DRIVE REGISTERS
010552  042767  000040  170406                  BIC     #BITS,ONCEE             ;CLEAR CLOCK COUNT FLAG
010560  104416                                  MROMD                           ;PUT DRIVE INTO MAINT MODE
010562  104420                                  MRCK                            ;CHECK MR REG TO
010564  022701                                  22701                           ;EQUAL 22701
010566  104424                                  MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
010570  012777  000001  170312                  MOV     #1,@RSDA                ;LOAD RSDA
010576  012777  000031  170274                  MOV     #31,@RSCS1              ;LOAD BLOCK SEARCH FUNCTION
010604  104426                                  DSCK                            ;CHECK RSDS
010606  030400                                  30400                           ;TO EQUAL 30400
010610  104000                                  HLT                             ;DRY IN RSCS SHOULD BE
                                                                                ;CLEARED FOR DRIVE WAS
                                                                                ;ISSUED A BLOCK SEARCH FUNCTION
                                                                                ;RSDS=BAD GOOD=CORRECT ANS
010612  012777  177777  170274                  MOV     #-1,@RSER               ;MODIFY RSER
                                                                                ;THIS SHOULD CAUSE AN RMR
                                                                                ;ERROR FOR DRIVE WAS BUSY
                                                                                ;WHEN COMMAND WAS GIVEN
010620  017700  170270                          MOV     @RSER,BAD               ;GET RSER REG
010624  012701  000004                          MOV     #4,GOOD                 ;GET CORRECT ANS
010630  020100                                  CMP     GOOD,BAD                ;DID RMR SET IN RSER?
010632  001410                                  BEQ     1$                      ;YES
010634  104402  021435                          TYPE    .TRMR
010640  104402  010644                          TYPE    ,.+2                    ;.ASCIZ "RSER"
010652  104000                                  HLT                             ;RSER=BAD GOOD=CORRECT ANS
010654  104426                          1$:     DSCK                            ;CHECK RSDS TO
010656  150600                                  150600                          ;EQUAL 150600
010660  104000                                  HLT                             ;RSDS=BAD GOOD=CORRECT ANS
010662  022777  104230  170210                  CMP     #104230,@RSCS1          ;DID CORRECT BITS SET IN RSCS1
010670  001401                                  BEQ     4$                      ;YES
010672  104040                                  HLT     !DS                     ;RSCS1 SHOULD=104230
                                                                                ;RSDS SHOULD=150600
                                                                                ;RSER SHOULD=4
010674  104414                          4$:     CLROK                           ;CLEAR ALL REGISTERS
010676  005777  170212                          TST     @RSER                   ;RSER SHOULD CLEAR
010702  001401                                  BEQ     3$                      ;RSER OK
010704  104040                                  HLT     !DS                     ;RSER SHOULD=0 FOR THE
                                                                                ;CLEAR BIT WAS LOADED IN RSCS2
010706  022777  004200  170164          3$:     CMP     #4200,@RSCS1            ;RSCS1 SHOULD=4200 FOR THE
010714  001401                                  BEQ     .+4                     ;CLEAR BIT WAS LOADED IN RSCS2
010716  104040                                  HLT     !DS                     ;RSCS1 SHOULD=4200
```

```
                                      ;*************************************************************
                                      ;TEST 47                 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)
                                      ;*******************************************************  *************
010720  104400                        TST47:  SCOPE

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

010722  104414                        RMRC4:  CLRDK                         ;CLEAR ALL DRIVE REGISTERS
010724  104416                                MROMD                         ;PUT DRIVE INTO MAINT MODE
010726  104420                                MRCK                          ;CHECK MR REG TO
010730  022701                                22701                         ;EQUAL 22701
010732  104424                                MRINT                         ;INIT MAINT MODE (CLEAR MRSP)
010734  012777  000001  170146                MOV     #1,@RSDA              ;LOAD RSDA
010742  012777  000031  170130                MOV     #31,@RSCS1            ;LOAD BLOCK SEARCH FUNCTION
010750  104426                                DSCK                          ;CHECK RSDS
010752  030400                                30400                         ;TO EQUAL 30400
010754  104000                                HLT                           ;DRY IN RSDS SHOULD BE
                                                                            ;CLEARED FOR DRIVE WAS
                                                                            ;ISSUED A BLOCK SEARCH FUNCTION
                                                                            ;RSDS=BAD GOOD=CORRECT ANS
010756  005077  170134                        CLR     @RSAS                 ;WRITE INTO ATTENTION SUMMARY REGISTER.
                                                                            ;SHOULD BE NO RMR ERROR BECAUSE
                                                                            ;WRITING RSAS IS ALLOWED ANYTIME.
010762  017700  170126                        MOV     @RSER,BAD             ;GET RSER REG
010766  012701  000000                        MOV     #0,GOOD               ;GET CORRECT ANS
010772  020100                                CMP     GOOD,BAD              ;DID RMR SET IN RSER?
010774  001435                                BEQ     1$                    ;NO
010776  104402  011002                        TYPE    ,.+2                  ;.ASCIZ <15><12>"RMR ERROR SHOULD NOT SET WHILE WRITING
011066  104000                                HLT                           ;RSER=BAD GOOD=CORRECT ANS
011070  104426                        1$:     DSCK                          ;CHECK RSDS TO
011072  030400                                30400                         ;EQUAL 30400
011074  104000                                HLT                           ;RSDS=BAD GOOD=CORRECT ANS
011076  022777  004231  167774                CMP     #4231,@RSCS1          ;DID CORRECT BITS SET IN RSCS1
011104  001401                                BEQ     4$                    ;YES
011106  104040                                HLT     !DS                   ;RSCS1 SHOULD=4231
                                                                            ;RSDS SHOULD=30400
                                                                            ;RSER SHOULD=0
011110  104414                        4$:     CLRDK                         ;CLEAR ALL REGISTERS
011112  005777  167776                        TST     @RSER                 ;RSER SHOULD CLEAR
011116  001401                                BEQ     3$                    ;RSER OK
011120  104040                                HLT     !DS                   ;RSER SHOULD=0 FOR THE
                                                                            ;CLEAR BIT WAS LOADED IN RSCS2
011122  022777  004200  167750        3$:     CMP     #4200,@RSCS1          ;RSCS1 SHOULD=4200 FOR THE
011130  001401                                BEQ     .+4                   ;CLEAR BIT WAS LOADED IN RSCS2
011132  104040                                HLT     !DS                   ;RSCS1 SHOULD=4200
```

```
                                        ;*******************************************************************
                                        ;TEST 50                DRIVE SELECT TEST
                                        ;*******************************************************************
 011134  104400                         TSTSD:  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

 011136  104414                         MRDSEL: CLROK                   ;CLEAR ALL REGISTERS
 011140  104416                                 MRDMD                   ;PUT DRIVE INTO MAINT MODE
 011142  104420                                 MRCK                    ;CHECK MAINT REG
 011144  022701                                 22701                   ;TO EQUAL 22701
 011146  104424                                 MRINT                   ;INITIALIZE MAINT MODE (CLEAR MRSP)
                                                                        ;BY SENDING 2 CLOCK PULSES
 011150  012777  177777  167732                 MOV     #-1,@RSDA       ;LOAD DISK ADDR REG OF DRIVE UNDER TEST
                                        ;SEARCH FOR NON EXISTENT DRIVES
 011156  012767  000401  170030                 MOV     #401,WORK
 011164  005001                                 CLR     GOOD
 011166  010177  167710         1$:             MOV     GOOD,@RSCS2     ;LOAD UNIT NO
 011172  005777  167716                         TST     @RSER           ;IS THIS A NED?
 011176  032777  010000  167676                 BIT     #BIT12,@RSCS2   ;CHECK
 011204  001005                                 BNE     2$              ;YES
 011206  005201                                 INC     GOOD            ;UPDATE UNIT NUMBER
 011210  006167  170000                         ROL     WORK            ;KEEP LOOKING FOR NED
 011214  103460                                 BCS     NEDDON          ;COULD NOT FIND ANY NON EXISTENT DRIVES
 011216  000763                                 BR      1$              ;LOOK FOR NED
 011220  012777  004000  167652 2$:             MOV     #4000,@RSCS1    ;CLEAR NED
 011226  010167  167766                         MOV     GOOD,WORK1      ;SAVE NED NUMBER
 011232  010177  167644                         MOV     GOOD,@RSCS2     ;LOAD UNIT # OF NED INTO RSCS2
 011236  005077  167646                         CLR     @RSDA           ;WRITE INTO A NON EXISTENT DRIVE REG
                                                                        ;THIS SHOULD CAUSE NED TO
                                                                        ;SET IN RSCS2
 011242  017700  167634                         MOV     @RSCS2,BAD      ;GET RSCS2
 011246  052701  010100                         BIS     #10100,GOOD     ;PUT CORRECT ANS IN GOOD
                                                                        ;BY SETTING NED AND IR
 011252  020100                                 CMP     GOOD,BAD        ;IS RSCS2 CORRECT?
 011254  001401                                 BEQ     .+4             ;YES
 011256  104000                                 HLT                     ;RSCS2=BAD GOOD=CORRECT ANS

 011260  022777  160200  167612                 CMP     #160200,@RSCS1  ;IS CS1 CORRECT
 011266  001401                                 BEQ     .+4             ;YES
 011270  104004                                 HLT     !DA             ;TRE SHOULD BE SET IN CS1 BECAUSE
                                                                        ;OF NED ERROR IN RSCS2
                                                                        ;RSCS1 SHOULD=160200
```

```
011272  005777  167620                      TST     @RSAS           ;DID ANY ATTENTION BITS SET?
011276  001401                              BEQ     .+4             ;NO
011300  104100                              HLT     !AS             ;NO ATTENTION BITS SHOULD BE SET
011302  112777  000100  167624              MOVB    #100,@RSCS1B    ;CLEAR TRE
011310  032777  010000  167564              BIT     #NED,@RSCS2     ;DID NED CLEAR
011316  001401                              BEQ     .+4             ;YES
011320  104040                              HLT     !DS             ;NED DID NOT CLEAR IN RSCS2
                                                                    ;BY CLEARING TRE BIT IN RSCS1
011322  016777  167632  167552              MOV     UNNUM,@RSCS2    ;LOAD CORRECT UNIT NUMBER
011330  022777  177777  167552              CMP     #-1,@RSDA       ;DID RSDA GET MODIFIED
                                                                    ;WHILE WRITING INTO A NON
                                                                    ;EXISTENT DRIVE?
011336  001443                              BEQ     NNDO            ;NO
011340  104004                              HLT     !DA             ;RSDA SHOULD= -1
011342  016700  167652                      MOV     WORK1,BAD       ;IT GOT MODIFIED WHILE WRITING
011346  016701  167606                      MOV     UNNUM,GOOD      ;INTO A NED
011352  104000                              HLT                     ;GOOD=DRIVE UNDER TEST
011354  000434                              BR      NNDO            ;BAD=NON EXISTENT DRIVE THAT WAS
                                                                    ;IN RSCS2 WHEN RSDA GOT MODIFIED
011356  032767  010000  167602  NEDDON:     BIT     #BIT12,ONCEE    ;WAS THIS TYPED BEFORE?
011364  001030                              BNE     NNDO            ;YES
011366  104402  011372                      TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
011440  052767  010000  167520              BIS     #BIT12,ONCEE    ;SET TYPED MESSAGE FLAG
011446                          NNDO:
```

```
                                                ;*******************************************************************
                                                ;TEST 51                 MAINTENANCE MODE WRITE TEST
                                                ;*******************************************************************
011446  104400                                  TST51:  SCOPE

                                                ;MODULE TESTED: M7771, M7753, M7751
                                                ;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.

011450  012767  000002  167466                          MOV     #2,FLAG2                ;SET TEST FLAG
011456  104414                                  MRWRT:  CLROK                           ;CLEAR DRIVE REGISTERS
011460  052767  000040  167500                          BIS     #BIT5,ONCEE             ;SET TYPE CLOCK COUNT FLAG
011466  042767  000600  167472                          BIC     #600,ONCEE              ;CLEAF FLAG BITS
011474  104430                                          MRIND                           ;SEND INDEX PULSE TO MR RE;
011476  104420                                          MRCK                            ;CHE.K MR REG
011500  022701                                          22701                           ;TO EQUAL 22701
011502  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)

011504  012702  026610                                  MOV     #INBUF,R2               ;GET LOCATION OF OUTBUF
011510  005022                                          CLR     (R2)+                   ;CLEAR 1ST LOCATION
011512  012722  177777                                  MOV     #-1,(R2)+               ;2ND WORD OF ALL ONES
011516  005003                                          CLR     R3                      ;CLEAR WORK LOC TO GENERATE
011520  000261                                          SEC                             ;A PATTERN OF FLOATING ONES
011522  006103                                  1$:     ROL     R3                      ;GET PATTERN
011524  103402                                          BCS     2$                      ;DONE GET OUT
011526  010322                                          MOV     R3,(R2)+                ;FILL BUFFER
011530  000774                                          BR      1$                      ;CONT
011532  012703  000156                          2$:     MOV     #110.,R3                ;FILL REMAINING PORTION OF

011536  012704  146314                                  MOV     #146314,R4              ;BUFFER WITH A PATTERN OF 146314
011542  010422                                  3$:     MOV     R4,(R2)+                ;LOAD BUFFER
011544  005303                                          DEC     R3                      ;DONE YET?
011546  001375                                          BNE     3$                      ;NO

                                                ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) TO SECTOR 0
011550  012777  026610  167330                          MOV     #INBUF,@RSBA            ;L.AD BUS ADDR REG
011556  012777  177600  167320                          MOV     #177600,@RSWC           ;LOAD WORD COUNT REG
011564  012777  000061  167306                          MOV     #61,@RSCS1              ;LOAD WRITE COMMAND
011572  104446                                          GETSP                           ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
011574  104220                                          HLT     !MR                     ;COULD NOT SET SECTOR PULSE (0)
011576  104450                                          SPASS                           ;CLOCK MR 2 TIMES SP = 1
```

```
                                            ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
011600  104430                                      MRIND
011602  104420                                      MRCK                        ;CHECK MR REG TO EQUAL
011604  020501                                      20501                       ;20501 FOR A
011606  104000                                      HLT                         ;WRITE COMD HAS BEEN ISSUED

                                            ;STEP THRU RESYNC PERIOD

011610  012767  001000  167364                      MOV     #512.,REPT
011616  052767  000040  167342                      BIS     #BITS,ONCEE         ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
011624  104422                              MRWRT1: MRCLK                       ;CLOCK MR REG
011626  104420                                      MRCK                        ;CHECK FOR
011630  070501                                      70501                       ;CORRECT DATA
011632  104000                                      HLT                         ;MR = BAD GOOD = CORRECT DATA
011634  104422                                      MRCLK                       ;CLOCK MR REG
011636  104420                                      MRCK                        ;CHECK FOR
011640  020501                                      20501                       ;CORRECT DATA
011642  104000                                      HLT                         ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
011644  005367  167332                              DEC     REPT                ;FINISH LOOPING
011650  001365                                      BNE     MRWRT1              ;THROUGH RESYNC PERIOD

                                            ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                            ;SP=0 EQUALS SECTOR PULSE
011652  104422                                      MRCLK                       ;CLOCK MR REG
011654  104420                                      MRCK                        ;MR SHOULD
011656  070101                                      70101                       ;EQUAL 70101
011660  104000                                      HLT                         ;MR=BAD GOOD=CORRECT ANS
011662  104422                                      MRCLK                       ;CLOCK MR REG
011664  104420                                      MRCK                        ;CHECK MR
011666  020101                                      20101                       ;TO EQUAL 20101
011670  104000                                      HLT                         ;MR=BAD GOOD=CORRECT ANS

                                            ;PERFORM 63 DOUBLE MAINT CLOCK OPERATIONS--WRITING PREAMBLE

011672  012767  000077  167302                      MOV     #63.,REPT
011700  104422                              MRWRT2: MRCLK                       ;CLOCK MR REG
011702  104420                                      MRCK                        ;CHECK MR REG
011704  071501                                      71501                       ;TO EQUAL 71501
011706  104000                                      HLT                         ;MR=BAD GOOD=CORRECT ANS
011710  104422                                      MRCLK                       ;CLOCK MR REG
011712  104420                                      MRCK                        ;CHECK MR REG
011714  021501                                      21501                       ;TO EQUAL 21501
011716  104000                                      HLT                         ;MR=BAD GOOD=CORRECT ANS
011720  005367  167256                              DEC     REPT                ;DONE YET
011724  001365                                      BNE     MRWRT2              ;NO LOOP
```

```
                                      ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN

 011726  104422                                  MRCLK                        ;CLOCK MR REG
 011730  104420                                  MRCK                         ;CHECK MR REG
 011732  171501                                  171501                       ;TO EQUAL 171501
 011734  104000                                  HLT                          ;MR REG=BAD GOOD=CORRECT ANS
 011736  104422                                  MRCLK                        ;CLOCK MR REG
 011740  104420                                  MRCK                         ;MR REG SHOULD
 011742  025501                                  25501                        ;EQUAL 25501
 011744  104000                                  HLT                          ;MR REG=BAD GOOD=CORRECT ANS
 011746  104422                                  MRCLK
 011750  104420                                  MRCK
 011752  175501                                  175501
 011754  104000                                  HLT
                                      ;PERFORM NEXT STEP 3 TIMES TO FINISH WRITTING PREAMBLE
 011756  012767  000003  167216                  MOV     #3,REPT
 011764  104422                       MRWRT3:    MRCLK                        ;CLOCK MR REG
 011766  104420                                  MRCK                         ;CHECK MR REG
 011770  025501                                  25501                        ;TO EQUAL 25501
 011772  104000                                  HLT                          ;MR=BAD GOOD=CORRECT ANS
 011774  104422                                  MRCLK                        ;CLOCK MR REG
 011776  104420                                  MRCK                         ;CHECK MR REG
 012000  175501                                  175501                       ;TO EQUAL 175501
 012002  104000                                  HLT                          ;MR REG=BAD GOOD=CORRECT ANS
 012004  005367  167172                          DEC     REPT                 ;DONE YES?
 012010  001365                                  BNE     MRWRT3               ;NO LOOP BACK

                                      ;MOVE DATA WORD INTO RS04 SHIFT REGISTER (M7753)

 012012  104422                                  MRCLK                        ;CLOCK MR REG
 012014  104420                                  MRCK                         ;CHECK MR REG
 012016  027501                                  27501                        ;TO EQUAL 27501
 012020  104000                                  HLT                          ;MR=BAD GOOD=CORRECT ANS
 012022  104422                                  MRCLK                        ;CLOCK MR REG
 012024  104420                                  MRCK                         ;MR REG SHOULD
 012026  123501                                  123501                       ;EQUAL 123501
 012030  104000                                  HLT                          ;MR=BAD GOOD=CORRECT ANS

                                      ;ENCODE SYNC 1 (M7751)

 012032  104422                                  MRCLK                        ;CLOCK MR REG
 012034  104420                                  MRCK                         ;MR REG SHOULD NOW
 012036  073501                                  73501                        ;EQUAL 73501
 012040  104000                                  HLT                          ;MR=BAD GOOD=CORRECT ANS
 012042  012705  026610                          MOV     #INBUF,R5            ;GET STARTING ADDR FOR DATA BUFFER
 012046  011504                                  MOV     (R5),R4              ;GET DATA
```

```
012050  012767  002167  167136           MOV     #1143.,WORK          ;DOING A 1 SECTOR TRANSFER 127 WORDS
                                                                      ;18 BITS PER WORD-CLOCK LOOPS
                                                                      ;TAKE CARE OF 2 BITS AT A TIME
                                                                      ;127 TIMES 9 EQUALS 1143 LOOPS
                                                                      ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
012056  042767  000200  167102           BIC     #BIT7,ONCEE          ;CLEAR LAST WORD FLAG
012064  052767  000100  167074           BIS     #BIT6,ONCEE          ;SET 1ST TRANSFER WORD FLAG
012072  104432                   1$:     Y9IT                         ;GET 2 BITS OF DATA
012074  104434                           CLKD1                        ;SEND FIRST CLOCK PULSE
                                                                      ;AND CALCULATE MR REG
                                                                      ;FOR CORRECT DATA (MWDT+MWDB)
012076  104000                           HLT                          ;MR REG NOT CORRECT
012100  104436                           CLKD2                        ;SEND 2ND CLOCK PULSE TO
                                                                      ;COMPLETE TRANSFER OF 2 BITS
                                                                      ;CALCULATE CORRECT ANS FOR
                                                                      ;MR REG (MWDT+MWDB)
012102  104000                           HLT                          ;MR=BAD GOOD=CORRECT ANS
012104  032767  000200  167054           BIT     #BIT7,ONCEE          ;ON LAST WORD YET?
012112  001015                           BNE     2$                   ;YES
012114  032767  000400  167044           BIT     #BIT8,ONCEE          ;ON CRC WORD YET?
012122  001040                           BNE     3$                   ;YES
012124  005367  167064                   DEC     WORK                 ;DONE WITH 127 WORDS?
012130  001360                           BNE     1$                   ;NO

012132  052767  000200  167026           BIS     #BIT7,ONCEE          ;SET LAST WORD FLAG
012140  012767  000012  167046           MOV     #10.,WORK            ;SET UP TO TRANSFER LAST WORD
012146  005367  167042           2$:     DEC     WORK                 ;DONE YET?
012152  001347                           BNE     1$                   ;NO

012154  052767  000400  167004           BIS     #BIT8,ONCEE          ;SET TRANSFERRING CRC WORD
012162  042767  000200  166776           BIC     #BIT7,ONCEE          ;CLEAR LAST WORD FLAG
012170  004767  011262                   JSR     PC,GENCRC            ;GENERATE CRC WORD
                                                                      ;AND LEAVE IN "WORK"
012174  012702  025610                   MOV     #INBUF,R2            ;GO TO END
012200  062702  000400                   ADD     #400,R2              ;OF DATA BUFFER
012204  016712  167004                   MOV     WORK,@R2             ;LOAD CRC WORD
012210  010205                           MOV     R2,R5                ;RESET POINTER FOR
012212  162705  000002                   SUB     #2,R5                ;R5 FOR CRC WD
012216  012767  000012  166770           MOV     #10.,WORK            ;SETUP TO XFER CRC
012224  005367  166764           3$:     DEC     WORK                 ;DONE YET
012230  001320                           BNE     1$                   ;NO
```

```
                                                ;EBL SHOULD NOW ASSERT

012232  104422                                          MRCLK                   ;CLOCK MR REG TO STOP THROUGH
                                                                                ;THE RS04 DISK SECTOR DEAD BAND AREA
012234  104420                                          MRCK                    ;CHECK MR REG
012236  153501                                          153501                  ;TO EQUAL 103501
012240  104000                                          HLT                     ;MR REG=BAD GOOD=CORRECT ANS

                                                ;LOOP 6 TIMES

012242  012767  000006  166732                          MOV     #6,REPT
012250  104422                                  4$:     MRCLK                   ;CLOCK MR REG
012252  104420                                          MRCK                    ;CHECK MR REG
012254  003501                                          3501                    ;TO EQUAL 53501
012256  104000                                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012260  104422                                          MRCLK                   ;CLOCK MR REG
012262  104420                                          MRCK                    ;CHECK MR REG
012264  153501                                          153501                  ;TO EQUAL 153501
012266  104000                                          HLT                     ;MR=BAD GOOD=CORRECT ANS
012270  005367  166706                                  DEC     REPT            ;DONE LOOPING YET?
012274  001365                                          BNE     4$              ;NO

                                                ;FINISH UP

012276  104422                                          MRCLK                   ;CLOCK MR REG
012300  104420                                          MRCK                    ;CHECK MR REG
012302  003501                                          3501                    ;TO EQUAL 3501
012304  104000                                          HLT                     ;MR REG=BAD GOOD=CORRECT ANS
012306  104422                                          MRCLK                   ;CLOCK MR REG
012310  104420                                          MRCK                    ;CHECK MR REG
012312  151501                                          151501                  ;TO EQUAL 151501
012314  104000                                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                                ;TRANSFER SHOULD NOW BE COMPLETE

012316  104422                                          MRCLK                   ;CLOCK MR REG
012320  104420                                          MRCK                    ;CHECK MRE REG
012322  002701                                          2701                    ;TO EQUAL 2701
012324  104000                                          HLT                     ;MR=BAD GOOD=CORRECT ANS

                                                ;NOW TEST CONTROLLER

012326  005777  166546                                  TST     @RSCS1          ;ANY ERRORS?
012332  100001                                          BPL     5$              ;NO
012334  104014                                          HLT     !DA!WC          ;YES
012336  005777  166542                          5$:     TST     @RSWC           ;DID WC GO TO 0
012342  001401                                          BEQ     .+4             ;YES
012344  104010                                          HLT     !WC             ;WC SHOULD BE = TO 0
012346  022777  000001  166534                          CMP     #1,@RSDA        ;DOES RSDA=1
012354  001401                                          BEQ     .+4             ;YES
012356  104004                                          HLT     !DA             ;RSDA SHOULD=1
012360  032767  000002  166556                          BIT     #BIT1,FLAG2     ;IN MAINT VERIFY TEST
012366  001002                                          BNE     .+6             ;NO
012370  000137  020720                                  JMP     @#MRVR2         ;YES, GO TO VERIFY TEST
```

```
                                        ;*************************************************************************
                                        ;TEST 52                  MAINTENANCE READ TEST
                                        ;*************************************************************************
012374  104400                          TST52:  SCOPE

                                        ;MODULE TESTED: M7771, M7753, M7751
                                        ;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.)

012376  104414                          MRRD:   CLRDK                           ;CLEAR DRIVE REGISTERS
012400  052767  000040  166560                  BIS     #BITS,ONCEE             ;SET TYPE CLOCK COUNT FLAG
012406  042767  147716  166552                  BIC     #147716,ONCEE           ;CLEAR ALL OTHER FLAG BITS
012414  104430                                  MRIND                           ;SEND INDEX PULSE TO MR REG
012416  104420                                  MRCK                            ;CHECK MR REG
012420  022701                                  22701                           ;TO EQUAL 22701
012422  104424                                  MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                                ;BY SENDING 2 CLOCK PULSES

012424  005067  166514                          CLR     FLAG2                   ;CLEAR FLAG TEST BITS

                                        ;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)

012430  012702  026610                          MOV     #INBUF,R2               ;GET LOCATION OF INBUF
012434  005022                                  CLR     (R2)+                   ;CLEAR 1ST LOCATION
012436  012722  177777                          MOV     #-1,(R2)+               ;2ND WORD OF ALL ONES
012442  005003                                  CLR     R3                      ;CLEAR WORK LOC TO GENERATE
012444  000261                                  SEC                             ;A PATTERN OF FLOATING ONES
012446  006103                          1$:     ROL     R3                      ;GET PATTERN
012450  103402                                  BCS     2$                      ;DONE GET OUT
012452  010322                                  MOV     R3,(R2)+                ;FILL BUFFER
012454  000774                                  BR      1$                      ;CONT
012456  012703  000156                  2$:     MOV     #110.,R3                ;FILL REMAINING PORTION OF
012462  012704  146314                          MOV     #146314,R4              ;BUFFER WITH A PATTERN OF 146314
012466  010422                          3$:     MOV     R4,(R2)+                ;LOAD BUFFER
012470  005303                                  DEC     R3                      ;DONE YET
012472  001375                                  BNE     3$                      ;NO
                                        ;NOTE:
                                        ;INBUF CONTAINS THE TABLE OF DATA WHICH IS "READ".
                                        ;VIA THE MRDB AND MRDT BITS IN RSMR
                                        ;OUTBUF IS WHERE THE DATA WORDS FROM THE
                                        ;MASSBUS ARE STORED
                                        ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0

 012474  012777  027410  166404                 MOV     #OUTBUF,@RSBA   ;LOAD BUSS ADDR REG
 012502  012777  177600  166374                 MOV     #177600,@RSWC   ;LOAD WORD COUNT REG
 012510  012777  000071  166362                 MOV     #71,@RSCS1      ;LOAD READ COMMAND
 012516  012702  000200                         MOV     #200,R2         ;CLEAR THE OUTBUF TABLE
 012522  012703  027410                         MOV     #OUTBUF,R3      ;SO THAT WHEN THE READ
 012526  005023                         4$:     CLR     (R3)+           ;IS FINISHED, WE CAN COMPARE
 012530  005302                                 DEC     R2              ;WHAT WE GOT (OUTBUF) WITH WHAT
 012532  001375                                 BNE     4$              ;WE EXPECTED (INBUF).
 012534  104446                                 GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
 012536  104220                                 HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
 012540  104450                                 SPASS                   ;CLOCK MR REG 2 TIMES SP = 1

                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
 012542  104430                                 MRIND
 012544  104420                                 MRCK                    ;CHECK MR REG TO EQUAL
 012546  022601                                 22601                   ;22601 FOR A
 012550  104000                                 HLT                     ;READ COMD

                                        ;STEP THRU RESYNC PERIOD

 012552  012767  001000  166422                 MOV     #512.,REPT
 012560  052767  000040  166400                 BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERRORS OCCUR
 012566  104422                         MRRD1:  MRCLK                   ;CLOCK MR REG
 012570  104420                                 MRCK                    ;CHECK FOR
 012572  072601                                 72601                   ;CORRECT DATA
 012574  104000                                 HLT                     ;MR=BAD GOOD=CORRECT DATA
 012576  104422                                 MRCLK                   ;CLOCK MR REG
 012600  104420                                 MRCK                    ;CHECK FOR
 012602  022601                                 22601                   ;CORRECT DATA
 012604  104000                                 HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
 012606  005367  166370                         DEC     REPT            ;FINISH LOOPING
 012612  001365                                 BNE     MRRD1           ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
 012614  104422                                 MRCLK                   ;CLOCK MR REG
 012616  104420                                 MRCK                    ;MR SHOULD
 012620  072201                                 72201                   ;EQUAL 72201
 012622  104000                                 HLT                     ;MR=BAD GOOD=CORRECT ANS
 012624  104422                                 MRCLK                   ;CLOCK MR REG
 012626  104420                                 MRCK                    ;CHECK MR
 012630  022201                                 22201                   ;TO EQUAL 22201
 012632  104000                                 HLT                     ;MR=BAD GOOD=CORRECT ANS
```

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

012634  012767  000107  166340            MOV     #71.,REPT
012642  104422                  MRRD2:    MRCLK                     ;CLOCK MR REG
012644  104420                            MRCK                      ;CHECK MR REG
012646  073601                            73601                     ;TO EQUAL 73601
012650  104000                            HLT                       ;MR=BAD GOOD=CORRECT ANS
012652  104422                            MRCLK                     ;CLOCK MR REG
012654  104420                            MRCK                      ;CHECK MR REG
012656  023601                            23601                     ;TO EQUAL 23601
012660  104000                            HLT                       ;MR=BAD GOOD=CORRECT ANS
012662  005367  166314                    DEC     REPT              ;DONE YET
012666  001365                            BNE     MRRD2             ;NO LOOP
012670  104422                            MRCLK                     ;CLOCK MR REG
012672  104420                            MRCK                      ;CHECK MR REG
012674  073601                            73601                     ;TO EQUAL 73601
012676  104000                            HLT                       ;MR=BAD GOOD=CORRECT ANS

                                ;READ SYNC"1"

012700  012777  000055  166216            MOV     #55,@RSMR
012706  012777  000045  166210            MOV     #45,@RSMR
012714  104420                            MRCK                      ;CHECK MR REG
012716  023645                            23645                     ;TO EQUAL
012720  104000                            HLT                       ;CONTENTS OF GOOD
012722  012777  000055  166174            MOV     #55,@RSMR
012730  012777  000045  166166            MOV     #45,@RSMR
012736  104420                            MRCK
012740  173645                            173645
012742  104000                            HLT

                                ;READ DATA
012744  005067  166254          MRD3:     CLR     WORK3             ;CLEAR CLOCK COUNT FOR DATA WD
012750  012705  026610                    MOV     #INBUF,R5         ;GET STARTING ADDRESS FOR DATA BUFFER
012754  162705  000002                    SUB     #2,R5
012760  012767  000025  166216            MOV     #21.,REPT1        ;SETUP COUNTER FOR 1ST SB BIT
012766  012767  002200  166206            MOV     #1152.,REPT       ;SETUP COUNTER TO TRANSFER
                                                                    ;128 WORDS-9X128=1152
                                                                    ;2 CLOCKS PER 2 BITS OF DATA
012774  104444                  1$:       RBIT                      ;GET 2 DATA BITS
012776  104440                            CLKR1                     ;CLOCK MR
013000  104000                            HLT                       ;MR REG NOT CORRECT
013002  104442                            CLKR2                     ;CLOCK MR REG
013004  104000                            HLT                       ;MR REG NOT CORRECT
013006  005367  166170                    DEC     REPT              ;DONE WITH DATA BUFFER YET?
013012  001370                            BNE     1$                ;NO
```

```
013014  032767  000400  166144  2$:     BIT     #BIT8,ONCEE     ;DID WE ALREADY DO CRC?
013022  001030                          BNE     3$              ;YES
013024  052767  000400  166134          BIS     #BIT8,ONCEE     ;NO SET CRC FLAG
013032  016767  166146  166132          MOV     REPT1,SAVEE     ;SAVE REPT1
013040  004767  010412                  JSR     PC,GENCRC       ;GENERATE CRC WORD
                                                                ;AND LEAVE IN LOC "WORK"
013044  012702  026610                  MOV     #INBUF,R2       ;
013050  016767  166116  166126          MOV     SAVEE,REPT1     ;RESTORE REPT1
013056  062702  000400                  ADD     #400,R2         ;STORE CRC WORD AT END OF
013062  016712  166126                  MOV     WORK,@R2        ;INBUF TABLE
013066  010205                          MOV     R2,R5
013070  162705  000002                  SUB     #2,R5
013074  012767  000011  166100          MOV     #9.,REPT        ;SETUP TO TRANSFER 1 WD
013102  000734                          BR      1$              ;TRANSFER CRC WD
013104  104422                  3$:     MRCLK                   ;CLOCK MR REG
013106  104420                          MRCK                    ;CHECK MR REG
013110  003601                          3601                    ;TO EQUAL
013112  104000                          HLT                     ;3601
013114  104422                          MRCLK                   ;CLOCK MR REG
013116  104420                          MRCK                    ;CHECK MR
013120  153601                          153601                  ;TO EQUAL
013122  104000                          HLT                     ;153601
013124  104422                          MRCLK                   ;CLOCK MR REG
013126  104420                          MRCK                    ;CHECK MR
013130  007601                          7601                    ;TO EQUAL
013132  104000                          HLT                     ;7601
013134  104422                          MRCLK                   ;CLOCK MR REG
013136  104420                          MRCK                    ;CHECK MR
013140  153601                          153601                  ;TO EQUAL
013142  104000                          HLT                     ;153601

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

013144  012767  000010  166030  MRD4:   MOV     #8.,REPT
013152  104422                  1$:     MRCLK                   ;CLOCK MR REG
013154  104420                          MRCK                    ;CHECK MR REG
013156  003601                          3601                    ;TO EQUAL
013160  104000                          HLT                     ;3601
013162  104422                          MRCLK                   ;CLOCK MR REG
013164  104420                          MRCK                    ;CHECK MR
013166  153601                          153601                  ;REG TO
013170  104000                          HLT                     ;EQUAL 153601
013172  005367  166004                  DEC     REPT            ;DONE YET?
013176  001365                          BNE     1$              ;NO

                                ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                ;SHOULD GET STROBE BUFFER

013200  104422                          MRCLK                   ;CLOCK MR REG
013202  104420                          MRCK                    ;CHECK MR
013204  007601                          7601                    ;REG TO
013206  104000                          HLT                     ;EQUAL 7601
```

```
                                          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                          ;SHOULD COMPLETE TRANSFER.

013210  104422                            MRD5:   MRCLK                        ;CLOCK MR REG
013212  022777  004270  165660                    CMP     #4270,@RSCS1         ;ANY ERRORS?
013220  001401                                    BEQ     1$                   ;NO
013222  104054                                    HLT     !DA!DS!WC
013224  005777  165654                    1$:     TST     @RSWC                ;DID WC GO TO 0
013230  001401                                    BEQ     .+4                  ;YES
013232  104010                                    HLT     !WC                  ;WC REG SHOULD=0
013234  022777  000001  165646                    CMP     #1,@RSDA             ;DOES RSDA=1
013242  001401                                    BEQ     .+4                  ;YES
013244  104004                                    HLT     !DA                  ;RSDA SHOULD=1

                                          ;COMPARE DATA READ WITH INPUT BUFFER
                                          ;WILL C'LY TYPEOUT 10 ERRORS --- BUT IF SW12 IS SET
                                          ;IT WILL TYPE OUT ALL ERRORS

013246  012700  026610                    MRD6:   MOV     #INBUF,BAD           ;GET STARTING LOC OF EXPECTED DATA
013252  012701  027410                            MOV     #OUTBUF,GOOD         ;GET STARTING LOC OF DATA "READ" FROM DISK
013256  012767  000012  165716                    MOV     #12,REPT             ;SET UP ERROR COUNTER
013264  012705  000201                            MOV     #201,RS              ;COMPARE 1 SECTOR
013270  005305                            3$:     DEC     RS                   ;DONE WITH SECTOR
013272  001433                                    BEQ     2$                   ;YES GET OUT
013274  022021                                    CMP     (BAD)+,(GOOD)+       ;IS DATA CORRECT?
013276  001774                                    BEQ     3$                   ;YES
013300  032777  010000  164262                    BIT     #BIT12,@SWR          ;TYPE ALL ERRORS?
013306  001003                                    BNE     1$                   ;YES
013310  005367  165666                            DEC     REPT                 ;TYPED OUT 10 ERRORS YET?
013314  001422                                    BEQ     2$                   ;YES GET OUT
013316  024041                            1$:     CMP     -(BAD),-(GOOD)       ;GET ERROR
013320  104000                                    HLT                          ;TYPE OUT ERROR
013322  010067  165666                            MOV     BAD,WORK
013326  104402  013332                            TYPE    ,.+2                 ;.ASCIZ "BAD ADDRESS= "
013350  016746  165640                            MOV     WORK,-(6)            ;PUT WORK ON STACK
013354  104406                                    TYPES                        ;TYPE STACK IN OCTAL - SUPRESS
013356  022021                                    CMP     (BAD)+,(GOOD)+
013360  000743                                    BR      3$
013362                                    2$:                                  ;DONE
```

```
                                        ;*******************************************************************
                                        ;TEST 53                  MAINTENANCE MODE DATA WRITE CHECK TEST
                                        ;*******************************************************************
013362  104400                          TST53:  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.

013364  104414                          MRWCK:  CLRDK                           ;CLEAR DRIVE REGISTERS
013366  052767  000040  165572                  BIS     #BITS,ONCEE             ;SET TYPE CLOCK COUNT FLAG
013374  042767  147716  165564                  BIC     #147716,ONCEE           ;CLEAR ALL OTHER FLAG BITS
013402  104430                                  MRIND                           ;SEND INDEX PULSE TO MR REG
013404  104420                                  MRCK                            ;CHECK MR REG
013406  022701                                  22701                           ;TO EQUAL 22701
013410  104424                                  MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                                ;BY SENDING 2 CLOCK PULSES

013412  012767  000004  165524                  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)

013420  012702  026610                          MOV     #INBUF,R2               ;GET LOCATION OF INBUF
013424  005022                                  CLR     (R2)+                   ;CLEAR 1ST LOCATION
013426  012722  177777                          MOV     #-1,(R2)+               ;2ND WORD OF ALL ONES
013432  005003                                  CLR     R3                      ;CLEAR WORK LOC TO GENERATE
013434  000261                                  SEC                             ;A PATTERN OF FLOATING ONES
013436  006103                          1$:     ROL     R3                      ;GET PATTERN
013440  103402                                  BCS     2$                      ;DONE GET OUT
013442  010322                                  MOV     R3,(R2)+                ;FILL BUFFER
013444  000774                                  BR      1$                      ;CONT
013446  012703  000156                  2$:     MOV     #110.,R3                ;FILL REMAINING PORTION OF
013452  012704  146314                          MOV     #146314,R4              ;BUFFER WITH A PATTERN OF 146314
013456  010422                          3$:     MOV     R4,(R2)+                ;LOAD BUFFER
013460  005303                                  DEC     R3                      ;DONE YET
013462  001375                                  BNE     3$                      ;NO

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

013464  012777  026610  165414                  MOV     #INBUF,@RSBA            ;LOAD BUS ADDR REG
013472  012777  177600  165404                  MOV     #177600,@RSWC           ;LOAD WORD COUNT REG
013500  012777  000051  165372                  MOV     #51,@RSCS1              ;LOAD WRITE CHECK COMMAND
013506  104446                                  GETSP                           ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                                ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                                ;THAT OTHERWISE COULD NOT BE CLEARED.
013510  104220                                  HLT     !MR                     ;COULD NOT SET SECTOR PULSE (0)
013512  104450                                  SPASS                           ;CLOCK MR SECTOR PULSE = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
013514  104430                                   MRIND
013516  104420                                   MRCK                        ;CHECK MR REG TO EQUAL
013520  022701                                   22701                       ;22701 FOR A
013522  104000                                   HLT                         ;WRITE CHECK COMMAND

                                        ;STEP THRU RESYNC PERIOD

013524  012767  001000  165450                   MOV      #512.,REPT
013532  052767  000040  165426                   BIS      #BITS,ONCEE        ;TYPE OUT CLOCK COUNT IF ERRORS OCCUR
013540  104422                          MRWCK1:  MRCLK                       ;CLOCK MR REG
013542  104420                                   MRCK                        ;CHECK FOR
013544  072701                                   72701                       ;CORRECT DATA
013546  104000                                   HLT                         ;MR=BAD GOOD=CORRECT DATA
013550  104422                                   MRCLK                       ;CLOCK MR REG
013552  104420                                   MRCK                        ;CHECK FOR
013554  022701                                   22701                       ;CORRECT DATA
013556  104000                                   HLT                         ;ERROR WHILE CLOCKING THROUGH RESYNC
013560  005367  165416                           DEC      REPT               ;FINISH LOOPING
013564  001365                                   BNE      MRWCK1             ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
013566  104422                                   MRCLK                       ;CLOCK MR REG
013570  104420                                   MRCK                        ;MR SHOULD
013572  072301                                   72301                       ;EQUAL 72301
013574  104000                                   HLT                         ;MR=BAD GOOD=CORRECT ANS
013576  104422                                   MRCLK                       ;CLOCK MR REG
013600  104420                                   MRCK                        ;CHECK MR
013602  022301                                   22301                       ;TO EQUAL 22301
013604  104000                                   HLT                         ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--

013606  012767  000107  165366                   MOV      #71.,REPT
013614  104422                          MRWCK2:  MRCLK                       ;CLOCK MR REG
013616  104420                                   MRCK                        ;CHECK MR REG
013620  073701                                   73701                       ;TO EQUAL 73701
013622  104000                                   HLT                         ;MR=BAD GOOD=CORRECT ANS
013624  104422                                   MRCLK                       ;CLOCK MR REG
013626  104420                                   MRCK                        ;CHECK MR REG
013630  023701                                   23701                       ;TO EQUAL 23701
013632  104000                                   HLT                         ;MR=BAD GOOD=CORRECT ANS
013634  005367  165342                           DEC      REPT               ;DONE YET
013640  001365                                   BNE      MRWCK2             ;NO LOOP
013642  104422                                   MRCLK                       ;CLOCK MR REG
013644  104420                                   MRCK                        ;CHECK MR REG
013646  073701                                   73701                       ;TO EQUAL 73701
013650  104000                                   HLT                         ;MR=BAD GOOD=CORRECT ANS
```

```
                                         ;READ SYNC"1"

013652  012777  000055  165244                   MOV     #55,@RSMR
013660  012777  000045  165236                   MOV     #45,@RSMR
013666  104420                                   MRCK                         ;CHECK MR FOR
013670  023745                                   23745                        ;FOR CORRECT
013672  104000                                   HLT                          ;ANS IS IN GOOD
013674  012777  000055  165222                   MOV     #55,@RSMR
013702  012777  000045  165214                   MOV     #45,@RSMR
013710  104420                                   MRCK
013712  173745                                   173745
013714  104000                                   HLT

                                         ;READ DATA
013716  005067  165302                   MRWCK3: CLR     WORK3                ;CLEAR CLOCK COUNT FOR DATA WD
013722  012705  026610                           MOV     #INBUF,R5            ;GET STARTING ADDRESS FOR DATA BUFFER
013726  162705  000002                           SUB     #2,R5
013732  012767  000025  165244                   MOV     #21.,REPT1           ;SETUP COUNTER FOR 1ST SB BIT
013740  012767  002200  165234                   MOV     #1152.,REPT          ;SETUP COUNTER TO TRANSFER
                                                                              ;128 WORDS-9X128=1152
                                                                              ;2 CLOCKS PER 2 BITS OF DATA
013746  104444                           1$:     RBIT                         ;GET 2 DATA BITS
013750  104440                                   CLKR1                        ;CLOCK MR
013752  104000                                   HLT                          ;MR REG NOT CORRECT
013754  104442                                   CLKR2                        ;CLOCK MR REG
013756  104000                                   HLT                          ;MR REG NOT CORRECT
013760  005367  165216                           DEC     REPT                 ;DONE WITH DATA BUFFER YET?
013764  001370                                   BNE     1$                   ;NO
013766  032767  000400  165172           2$:     BIT     #BIT8,ONCEE          ;DID WE ALREADY DO CRC?
013774  001030                                   BNE     3$                   ;YES
013776  052767  000400  165162                   BIS     #BIT8,ONCEE          ;NO SET CRC FLAG
014004  016767  165174  165160                   MOV     REPT1,SAVEE          ;SAVE REPT1
014012  004767  007440                           JSR     PC,GENCRC            ;GENERATE CRC WORD
                                                                              ;AND LEAVE IN LOC "WORK"
014016  012702  026610                           MOV     #INBUF,R2
014022  016767  165144  165154                   MOV     SAVEE,REPT1          ;RESTORE REPT1
014030  062702  000400                           ADD     #400,R2              ;STORE CRC WORD TO BE READ
014034  016712  165154                           MOV     WORK,@R2             ;AT END OF INBUF TABLE
014040  010205                                   MOV     R2,R5
014042  162705  000002                           SUB     #2,R5
014046  012767  000011  165126                   MOV     #9.,REPT             ;SETUP TO TRANSFER 1 WD
014054  000734                                   BR      1$                   ;TRANSFER CRC WD
```

```
014056  104422                            3$:     MRCLK                    ;CLOCK MR REG
014060  104420                                    MRCK                     ;CHECK MR REG
014062  003701                                    3701                     ;TO EQUAL
014064  104000                                    HLT                      ;3701
014066  104422                                    MRCLK                    ;CLOCK MR REG
014070  104420                                    MRCK                     ;CHECK MR
014072  153701                                    153701                   ;TO EQUAL
014074  104000                                    HLT                      ;153701
014076  104422                                    MRCLK                    ;CLOCK MR REG
014100  104420                                    MRCK                     ;CHECK MR
014102  007701                                    7701                     ;TO EQUAL
014104  104000                                    HLT                      ;7701
014106  104422                                    MRCLK                    ;CLOCK MR REG
014110  104420                                    MRCK                     ;CHECK MR
014112  153701                                    153701                   ;TO EQUAL
014114  104000                                    HLT                      ;153701

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

014116  012767  000010  165056            MRWCK4: MOV     #8.,REPT
014124  104422                            1$:     MRCLK                    ;CLOCK MR REG
014126  104420                                    MRCK                     ;CHECK MR REG
014130  003701                                    3701                     ;TO EQUAL
014132  104000                                    HLT                      ;3601
014134  104422                                    MRCLK                    ;CLOCK MR REG
014136  104420                                    MRCK                     ;CHECK MR
014140  153701                                    153701                   ;REG TO
014142  104000                                    HLT                      ;EQUAL 153601
014144  005367  165032                            DEC     REPT             ;DONE YET?
014150  001365                                    BNE     1$               ;NO

                                          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                          ;SHOULD GET STROBE BUFFER

014152  104422                                    MRCLK                    ;CLOCK MR REG
014154  104420                                    MRCK                     ;CHECK MR
014156  007701                                    7701                     ;REG TO
014160  104000                                    HLT                      ;EQUAL 7601

                                          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                          ;SHOULD COMPLETE TRANSFER.

014162  104422                            MRWCK5: MRCLK                    ;CLOCK MR REG
014164  022777  004250  164706                    CMP     #4250,@RSCS1     ;ANY ERRORS?
014172  001401                                    BEQ     1$               ;NO
014174  104054                                    HLT     !DA!DS!WC
014176  005777  164702                    1$:     TST     @RSWC            ;DID WC GO TO 0
014202  001401                                    BEQ     .+4              ;YES
014204  104010                                    HLT     !WC              ;WC REG SHOULD=0
014206  022777  000001  164674                    CMP     #1,@RSDA         ;DOES RSDA=1
014214  001401                                    BEQ     .+4              ;YES
014216  104004                                    HLT     !DA              ;RSDA SHOULD=1
```

```
                                        ;*******************************************************************
                                        ;TEST 54                 MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)
                                        ;*******************************************************************
 014220  104400                         TST54:  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.

 014222  012767  000040  164714                 MOV     #40,FLAG2               ;CLEAR TST FLAG
 014230  104414                         MRCRC:  CLRDK                           ;CLEAR DRIVE REGISTERS
 014232  052767  000040  164726                 BIS     #BIT5,ONCEE             ;TYPE CLOCK COUNT IF ERROR OCCURS
 014240  042767  147716  164720                 BIC     #147716,ONCEE           ;CLEAR ALL OTHER FLAG BITS
 014246  104430                                 MRIND                           ;SEND INDEX PULSE TO MR REG
 014250  104420                                 MRCK                            ;CHECK MR REG
 014252  022701                                 22701                           ;TO EQUAL 22701
 014254  104424                                 MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                                ;BY SENDING 2 CLOCK PULSES
 014256  032767  000020  164660                 BIT     #BIT4,FLAG2             ;FIRST TIME THROUGH
 014264  001023                                 BNE     3$                      ;NO
 014266  012767  000001  164676                 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.

 014274  012702  026610                         MOV     #INBUF,R2               ;GET LOCATION OF INBUF
 014300  012703  000016                         MOV     #14.,R3                 ;SETUP COUNTER
 014304  005022                         1$:     CLR     (R2)+                   ;TO CLEAR THE
 014306  005303                                 DEC     R3                      ;FIRST 14
 014310  001375                                 BNE     1$                      ;WORDS
 014312  012722  000012                         MOV     #12,(R2)+               ;LOAD A 12
 014316  012722  020000                         MOV     #20000,(R2)+            ;LOAD A 20000
 014322  012703  000200                         MOV     #128.,R3                ;SETUP COUNTER
 014326  005022                         2$:     CLR     (R2)+                   ;TO CLEAR THE
 014330  005303                                 DEC     R3                      ;REMAINING WORDS
 014332  001375                                 BNE     2$                      ;FOR THAT SECTOR
```

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

 014334  012777  027410  164544  3$:     MOV     #OUTBUF,@RSBA      ;LOAD BUS ADDR REG
 014342  012777  177600  164534          MOV     #177600,@RSWC      ;LOAD WORD COUNT REG
 014350  012777  000071  164522          MOV     #71,@RSCS1         ;LOAD READ COMMAND
 014356  012702  000200                  MOV     #200,R2
 014362  012703  027410                  MOV     #OUTBUF,R3
 014366  052767  000020  164550          BIS     #BIT4,FLAG2        ;SET 1ST TIME THROUGH FLAG
 014374  005023                  4$:     CLR     (R3)+
 014376  005302                          DEC     R2
 014400  001375                          BNE     4$
 014402  104446                          GETSP                      ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                    ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                    ;THAT OTHERWISE COULD NOT BE CLEARED.
 014404  104220                          HLT     !MR                ;COULD NOT SET SECTOR PULSE (0)
 014406  104450                          SPASS                      ;CLOCK MR REG SP = 1

                                 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
 014410  104430                          MRIND
 014412  104420                          MRCK                       ;CHECK MR REG TO EQUAL
 014414  022601                          22601                      ;22601 FOR A
 014416  104000                          HLT                        ;READ COMD

                                 ;STEP THRU RESYNC PERIOD

 014420  012767  001000  164554          MOV     #512.,REPT
 014426  052767  000040  164532          BIS     #BIT5,ONCEE        ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
 014434  104422                  MRCRC1: MRCLK                      ;CLOCK MR REG
 014436  104420                          MRCK                       ;CHECK FOR
 014440  072601                          72601                      ;CORRECT DATA
 014442  104000                          HLT                        ;MR=BAD GOOD=CORRECT DATA
 014444  104422                          MRCLK                      ;CLOCK MR REG
 014446  104420                          MRCK                       ;CHECK FOR
 014450  022601                          22601                      ;CORRECT DATA
 014452  104000                          HLT                        ;ERROR WHILE CLOCKING THROUGH RESYNC
 014454  005367  164522                  DEC     REPT               ;FINISH LOOPING
 014460  001365                          BNE     MRCRC1             ;THROUGH RESYNC PERIOD

                                 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                 ;SP=0 EQUALS SECTOR PULSE
 014462  104422                          MRCLK                      ;CLOCK MR REG
 014464  104420                          MRCK                       ;MR SHOULD
 014466  072201                          72201                      ;EQUAL 72201
 014470  104000                          HLT                        ;MR=BAD GOOD=CORRECT ANS
 014472  104422                          MRCLK                      ;CLOCK MR REG
 014474  104420                          MRCK                       ;CHECK MR
 014476  022201                          22201                      ;TO EQUAL 22201
 014500  104000                          HLT                        ;MR=BAD GOOD=CORRECT ANS
```

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

      014502  012767  000107  164472             MOV     #71.,REPT
      014510  104422                     MRCRC2: MRCLK                           ;CLOCK MR REG
      014512  104420                             MRCK                            ;CHECK MR REG
      014514  073601                             73601                           ;TO EQUAL 73601
      014516  104000                             HLT                             ;MR=BAD GOOD=CORRECT ANS
      014520  104422                             MRCLK                           ;CLOCK MR REG
      014522  104420                             MRCK                            ;CHECK MR REG
      014524  023601                             23601                           ;TO EQUAL 23601
      014526  104000                             HLT                             ;MR=BAD GOOD=CORRECT ANS
      014530  005367  164446                     DEC     REPT                    ;DONE YET
      014534  001365                             BNE     MRCRC2                  ;NO LOOP
      014536  104422                             MRCLK                           ;CLOCK MR REG
      014540  104420                             MRCK                            ;CHECK MR REG
      014542  073601                             73601                           ;TO EQUAL 73601
      014544  104000                             HLT                             ;MR=BAD GOOD=CORRECT ANS

                                         ;READ SYNC"1"

      014546  012777  000055  164350             MOV     #55.@RSMR
      014554  012777  000045  164342             MOV     #45,@RSMR
      014562  104420                             MRCK                            ;CHECK MR REG
      014564  023645                             23645                           ;FOR CORRECT
      014566  104000                             HLT                             ;ANS IS IN GOOD
      014570  012777  000055  164326             MOV     #55,@RSMR
      014576  012777  000045  164320             MOV     #45,@RSMR
      014604  104420                             MRCK
      014606  173645                             173645
      014610  104000                             HLT

                                         ;READ DATA
      014612  005067  164406              MRCRC3: CLR     WORK3                   ;CLEAR CLOCK COUNT FOR DATA WD
      014616  012705  026610                     MOV     #INBUF,R5               ;GET STARTING ADDRESS FOR DATA BUFFER
      014622  162705  000002                     SUB     #2,R5
      014626  012767  000025  164350             MOV     #21.,REPT1              ;SETUP COUNTER FOR 1ST SB BIT
      014634  012767  002200  164340             MOV     #1152.,REPT             ;SETUP COUNTER TO TRANSFER
                                                                                 ;128 WORDS-9X128=1152
                                                                                 ;2 CLOCKS PER 2 BITS OF DATA
      014642  104444                      1$:     RBIT                            ;GET 2 DATA BITS
      014644  104440                             CLKR1                           ;CLOCK MR
      014646  104000                             HLT                             ;MR REG NOT CORRECT
      014650  104442                             CLKR2                           ;CLOCK MR REG
      014652  104000                             HLT                             ;MR REG NOT CORRECT
      014654  005367  164322                     DEC     REPT                    ;DONE WITH DATA BUFFER YET?
      014660  001370                             BNE     1$                      ;NO
```

```
  3231  014662  032767  000400  164276  2$:     BIT     #BIT8,ONCEE      ;DID WE ALREADY DO CRC?
  3232  014670  001020                          BNE     3$               ;YES
  3233  014672  052767  000400  164266          BIS     #BIT8,ONCEE      ;NO SET CRC FLAG
  3234  014700  012702  026610                  MOV     #INBUF,R2        ;MOVE CRC
  3235  014704  062702  000440                  ADD     #440,R2          ;WORD TO END OF
  3236  014710  016712  164256          4$:     MOV     SAVEE,@R2        ;INBUF TABLE
  3237  014714  010205                  5$:     MOV     R2,R5            ;GET CRC WORD
  3238  014716  162705  000002                  SUB     #2,R5
  3239  014722  012767  000011  164252          MOV     #9.,REPT         ;SETUP TO TRANSFER 1 WD
  3240  014730  000744                          BR      1$               ;TRANSFER CRC WD
  3241  014732  104422                  3$:     MRCLK                    ;CLOCK MR REG
  3242  014734  104420                          MRCK                     ;CHECK MR REG
  3243  014736  003601                          3601                     ;TO EQUAL
  3244  014740  104000                          HLT                      ;3601
  3245  014742  104422                          MRCLK                    ;CLOCK MR REG
  3246  014744  104420                          MRCK                     ;CHECK MR
  3247  014746  153601                          153601                   ;TO EQUAL
  3248  014750  104000                          HLT                      ;153601
  3249  014752  104422                          MRCLK                    ;CLOCK MR REG
  3250  014754  104420                          MRCK                     ;CHECK MR
  3251  014756  007601                          7601                     ;TO EQUAL
  3252  014760  104000                          HLT                      ;7601
  3253  014762  104422                          MRCLK                    ;CLOCK MR REG
  3254  014764  104420                          MRCK                     ;CHECK MR
  3255  014766  153601                          153601                   ;TO EQUAL
  3256  014770  104000                          HLT                      ;153601
```

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

014772  012767  000010  164202  MRCRC4: MOV     #8.,REPT

015000  104422                  1$:     MRCLK                           ;CLOCK MR REG
015002  104420                          MRCK                            ;CHECK MR REG
015004  003601                          3601                            ;TO EQUAL
015006  104000                          HLT                             ;3601
015010  104422                          MRCLK                           ;CLOCK MR REG
015012  104420                          MRCK                            ;CHECK MR
015014  153601                          153601                          ;REG TO
015016  104000                          HLT                             ;EQUAL 153601
015020  005367  164156                  DEC     REPT                    ;DONE YET?
015024  001365                          BNE     1$                      ;NO

                                      ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                      ;SHOULD GET STROBE BUFFER

015026  104422                          MRCLK                           ;CLOCK MR REG
015030  104420                          MRCK                            ;CHECK MR
015032  007601                          7601                            ;REG TO
015034  104000                          HLT                             ;EQUAL 7601

                                      ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                      ;SHOULD COMPLETE TRANSFER.

015036  104422                  MRCRC5: MRCLK                           ;CLOCK MR REG
015040  022777  004270  164032          CMP     #4270,@RSCS1            ;ANY ERRORS?
015046  001401                          BEQ     1$                      ;NO
015050  104054                          HLT     !DA!DS!WC
015052  005777  164026          1$:     TST     @RSWC                   ;DID WC GO TO 0
015056  001401                          BEQ     .+4                     ;YES
015060  104010                          HLT     !WC                     ;WC REG SHOULD=0
015062  006167  164104                  ROL     SAVEE                   ;GET NEXT CRC WORD
015066  103404                          BCS     2$                      ;DONE - BRANCH
015070  004767  010506                  JSR     PC,MDATA                ;SHIFT DATA PATTERN
015074  000167  177130                  JMP     MRCRC                   ;RESTART TEST WITH NEW DATA PATTERN
015100                          2$:                                     ;DONE
```

```
                                ;*******************************************************************
                                ;TEST 55                    MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)
                                ;*******************************************************************
015100  104400                  TST55:  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.
015102  012767  000040  164034          MOV     #40,FLAG2               ;CLEAR TST FLAG
015110  104414                  MRDCK:  CLRDK                           ;CLEAR DRIVE REGISTERS
015112  052767  000040  164046          BIS     #BIT5,ONCEE             ;TYPE CLOCK COUNT IF ERROR OCCURS
015120  042767  147716  164040          BIC     #147716,ONCEE           ;CLEAR ALL OTHER FLAG BITS
015126  104430                          MRIND                           ;SEND INDEX PULSE TO MR REG
015130  104420                          MRCK                            ;CHECK MR REG
015132  022701                          22701                           ;TO EQUAL 22701
015134  104424                          MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
                                                                        ;BY SENDING 2 CLOCK PULSES
015136  032767  000020  164000          BIT     #BIT4,FLAG2             ;FIRST TIME THROUGH
015144  001023                          BNE     3$                      ;NO
015146  012767  000001  164016          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
015154  012702  026610                  MOV     #INBUF,R2               ;GET LOCATION OF OUTBUF
015160  012703  000017                  MOV     #15.,R3                 ;SETUP COUNTER
015164  005022                  1$:     CLR     (R2)+                   ;TO CLEAR THE
015166  005303                          DEC     R3                      ;FIRST 15
015170  001375                          BNE     1$                      ;WORDS
015172  012722  000001                  MOV     #1,(R2)+                ;LOAD A 1
015176  012722  042000                  MOV     #42000,(R2)+            ;LOAD A 42000
015202  012703  000177                  MOV     #127.,R3                ;SETUP COUNTER
015206  005022                  2$:     CLR     (R2)+                   ;TO CLEAR THE
015210  005303                          DEC     R3                      ;REMAINING WORDS
015212  001375                          BNE     2$                      ;FOR THAT SECTOR
                                ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
015214  012777  027410  163664  3$:     MOV     #OUTBUF,@RSBA           ;LOAD BUS ADDR REG
015222  012777  177600  163654          MOV     #177600,@RSWC           ;LOAD WORD COUNT REG
015230  012777  000071  163642          MOV     #71,@RSCS1              ;LOAD READ COMMAND
015236  012702  000200                  MOV     #200,R2
015242  012703  027410                  MOV     #OUTBUF,R3
015246  052767  000020  163670          BIS     #BIT4,FLAG2             ;SET 1ST TIME THROUGH FLAG
015254  005023                  4$:     CLR     (R3)+
015256  005302                          DEC     R2
015260  001375                          BNE     4$
015262  104446                          GETSP                           ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
015264  104220                          HLT     !MR                     ;COULD NOT SET SECTOR PULSE (0)
015266  104450                          SPASS                           ;CLOCK MR REG SP = 1
```

```
                                  ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
015270  104430                            MRIND
015272  104420                            MRCK                    ;CHECK MR REG TO EQUAL
015274  022601                            22601                   ;22601 FOR A
015276  104000                            HLT                     ;READ COMD

                                  ;STEP THRU RESYNC PERIOD

015300  012767  001000  163674            MOV     #512.,REPT
015306  052767  000040  163652            BIS     #BIT5,ONCEE     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
015314  104422                    MRDCK1: MRCLK                   ;CLOCK MR REG
015316  104420                            MRCK                    ;CHECK FOR
015320  072601                            72601                   ;CORRECT DATA
015322  104000                            HLT                     ;MR=BAD GOOD=CORRECT DATA
015324  104422                            MRCLK                   ;CLOCK MR REG
015326  104420                            MRCK                    ;CHECK FOR
015330  022601                            22601                   ;CORRECT DATA
015332  104000                            HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
015334  005367  163642                    DEC     REPT            ;FINISH LOOPING
015340  001365                            BNE     MRDCK1          ;THROUGH RESYNC PERIOD

                                  ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                  ;SP=0 EQUALS SECTOR PULSE
015342  104422                            MRCLK                   ;CLOCK MR REG
015344  104420                            MRCK                    ;MR SHOULD
015346  072201                            72201                   ;EQUAL 72201
015350  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
015352  104422                            MRCLK                   ;CLOCK MR REG
015354  104420                            MRCK                    ;CHECK MR
015356  022201                            22201                   ;TO EQUAL 22201
015360  104000                            HLT                     ;MR=BAD GOOD=CORRECT ANS
```

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

015362  012767  000107  163612                  MOV     #71.,REPT
015370  104422                          MRDCK2: MRCLK                           ;CLOCK MR REG
015372  104420                                  MRCK                            ;CHECK MR REG
015374  073601                                  73601                           ;TO EQUAL 73601
015376  104000                                  HLT                             ;MR=BAD GOOD=CORRECT ANS
015400  104422                                  MRCLK                           ;CLOCK MR REG
015402  104420                                  MRCK                            ;CHECK MR REG
015404  023601                                  23601                           ;TO EQUAL 23601
015406  104000                                  HLT                             ;MR=BAD GOOD=CORRECT ANS
015410  005367  163566                          DEC     REPT                    ;DONE YET
015414  001365                                  BNE     MRDCK2                  ;NO LOOP
015416  104422                                  MRCLK                           ;CLOCK MR REG
015420  104420                                  MRCK                            ;CHECK MR REG
015422  073601                                  73601                           ;TO EQUAL 73601
015424  104000                                  HLT                             ;MR=BAD GOOD=CORRECT ANS

                                        ;READ SYNC"1"

015426  012777  000055  163470                  MOV     #55,@RSMR
015434  012777  000045  163462                  MOV     #45,@RSMR
015442  104420                                  MRCK                            ;CHECK MR REG
015444  023645                                  23645                           ;TO EQUAL
015446  104000                                  HLT                             ;CORRECT ANS IN GOOD
015450  012777  000055  163446                  MOV     #55,@RSMR
015456  012777  000045  163440                  MOV     #45,@RSMR
015464  104420                                  MRCK
015466  173645                                  173645
015470  104000                                  HLT

                                        ;READ DATA
015472  005067  163526                  MRDCK3: CLR     WORK3                   ;CLEAR CLOCK COUNT FOR DATA WD
015476  012705  026610                          MOV     #INBUF,R5               ;GET STARTING ADDRESS FOR DATA BUFFER
015502  162705  000002                          SUB     #2,R5
015506  012767  000025  163470                  MOV     #21.,REPT1              ;SETUP COUNTER FOR 1ST SB BIT
015514  012767  002200  163460                  MOV     #1152.,REPT             ;SETUP COUNTER TO TRANSFER
                                                                                ;128 WORDS-9X128=1152
                                                                                ;2 CLOCKS PER 2 BITS OF DATA
015522  104444                          1$:     RBIT                            ;GET 2 DATA BITS
015524  104440                                  CLKR1                           ;CLOCK MR
015526  104000                                  HLT                             ;MR REG NOT CORRECT
015530  104442                                  CLKR2                           ;CLOCK MR REG
015532  104000                                  HLT                             ;MR REG NOT CORRECT
015534  005367  163442                          DEC     REPT                    ;DONE WITH DATA BUFFER YET?
015540  001370                                  BNE     1$                      ;NO
```

```
  3428  015542  032767  000400  163416  2$:     BIT     #BIT8,ONCEE         ;DID WE ALREADY DO CRC?
  3429  015550  001020                          BNE     3$                  ;YES
  3430  015552  052767  000400  163406          BIS     #BIT8,ONCEE         ;NO SET CRC FLAG
  3431  015560  012702  026610                  MOV     #INBUF,R2           ;MOVE CRC
  3432  015564  062702  000440                  ADD     #440,R2             ;WORD TO END OF
  3433  015570  012712  000000          4$:     MOV     #0,@R2              ;INBUF TABLE
  3434  015574  010205                  5$:     MOV     R2,R5               ;GET CRC WORD
  3435  015576  162705  000002                  SUB     #2,R5
  3436  015602  012767  000011  163372          MOV     #9.,REPT            ;SETUP TO TRANSFER 1 WD
  3437  015610  000744                          BR      1$                  ;TRANSFER CRC WD
  3438  015612  104422                  3$:     MRCLK                       ;CLOCK MR REG
  3439  015614  104420                          MRCK                        ;CHECK MR REG
  3440  015616  003601                          3601                        ;TO EQUAL
  3441  015620  104000                          HLT                         ;3601
  3442  015622  104422                          MRCLK                       ;CLOCK MR REG
  3443  015624  104420                          MRCK                        ;CHECK MR
  3444  015626  153601                          153601                      ;TO EQUAL
  3445  015630  104000                          HLT                         ;153601
  3446  015632  104422                          MRCLK                       ;CLOCK MR REG
  3447  015634  104420                          MRCK                        ;CHECK MR
  3448  015636  007601                          7601                        ;TO EQUAL
  3449  015640  104000                          HLT                         ;7601
  3450  015642  104422                          MRCLK                       ;CLOCK MR REG
  3451  015644  104420                          MRCK                        ;CHECK MR
  3452  015646  153601                          153601                      ;TO EQUAL
  3453  015650  104000                          HLT                         ;153601
```

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

015652  012767  000010  163322  MRDCK4: MOV     #8.,REPT

015660  104422                  1$:     MRCLK                   ;CLOCK MR REG
015662  104420                          MRCK                    ;CHECK MR REG
015664  003601                          3601                    ;TO EQUAL
015666  104000                          HLT                     ;3601
015670  104422                          MRCLK                   ;CLOCK MR REG
015672  104420                          MRCK                    ;CHECK MR
015674  153601                          153601                  ,REG TO
015676  104000                          HLT                     ;EQUAL 153601
015700  005367  163276                  DEC     REPT            ;DONE YET?
015704  001365                          BNE     1$              ;NO

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD GET STROBE BUFFER

015706  104422                          MRCLK                   ;CLOCK MR REG
015710  104420                          MRCK                    ;CHECK MR
015712  007601                          7601                    ;REG TO
015714  104000                          HLT                     ;EQUAL 7601

                                        ;PERFORM ONE MAINTENANCE CLOCK OPERATION
                                        ;SHOULD COMPLETE TRANSFER.

015716  104422                  MRDCK5: MRCLK                   ;CLOCK MR REG
015720  022777  144270  163152          CMP     #144270,@RSCS1  ;IS RSCS1 CORRECT?
015726  001401                          BEQ     1$              ;YES
015730  104054                          HLT     !DA!DS!WC
015732  005777  163146          1$:     TST     @RSWC           ;DID WC GO TO 0
015736  001401                          BEQ     .+4             ;YES
015740  104010                          HLT     !WC             ;WC REG SHOULD=0
015742  022777  100000  163144          CMP     #100000,@RSER   ;DID DCK SET?
015750  001417                          BEQ     3$              ;YES
015752  104050                          HLT     !DS!WC
015754  104402  015760                  TYPE    ,.+2            ;.ASCIZ <15><12>"DCK DID NOT SET "
016004  004767  004210                  JSR     PC,CRCTYP       ;GET IC THAT FAILED AND TYPE IT
016010  000241                  3$:     CLC
016012  006167  163154                  ROL     SAVEE           ;GET NEXT CRC WORD
016016  103404                          BCS     2$              ;DONE - BRANCH
016020  004767  007556                  JSR     PC,MDATA        ;SHIFT DATA PATTERN
016024  000167  177060                  JMP     MRDCK           ;RESTART TEST WITH NEW DATA PATTERN
016030                          2$:                             ;DONE
```

```
                                                ;*************************************************************************
                                                ;TEST 56                        IGNORE F   TION TEST
                                                ;*************************************************************************
016030  104400                                  TST56:  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.

016032  104414                                  MRIFT:  CLROK                           ;CLEAR ALL REGISTERS
016034  104430                                          MRIND                           ;SEND INDEX PULSE TO MR REG
016036  104420                                          MRCK                            ;CHECK MR REG
016040  022701                                          22701                           ;TO EQUAL 22701
016042  104424                                          MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
016044  012777  177777  163042                          MOV     #-1,@RSER               ;SET ERRORS
016052  016777  163104  163036                          MOV     UNITSV,@RSAS            ;CLEAR ATA BIT IN RSAS
                                                                                        ;AND ERROR BITS IN RSCS1
016060  012777  027410  163020                          MOV     #OUTBUF,@RSBA           ;LOAD RSBA
016066  012777  177777  163010                          MOV     #-1,@RSWC               ;LOAD RSWC
016074  012777  000071  162776                          MOV     #71,@RSCS1              ;LOAD READ FUNCTION
016102  032777  000001  162770                          BIT     #BIT0,@RSCS1            ;IS "GO" BIT ZERO?
016110  001401                                          BEQ     1$                      ;YES
016112  104140                                          HLT     !DS!AS                  ;"GO" BIT IN RSCS1 SHOULD NOT
                                                                                        ;LOAD IF ERRORS ARE PRESENT IN THE DRIVE
016114  012767  177777  163072          1$:             MOV     #177777,WORK            ;SETUP TIMEOUT FOR MXF ERROR
016122  005367  163066                  5$:             DEC     WORK
016126  000240                                          NOP
016130  000240                                          NOP
016132  001373                                          BNE     5$
016134  017700  162742                                  MOV     @RSCS2,BAD              ;CHECK RSCS2 FOR MXF
016140  012701  001100                                  MOV     #1100,GOOD              ;GET CORRECT ANS
016144  056701  163010                                  BIS     UNNUM,GOOD              ;FOR RSCS2
016150  020001                                          CMP     BAD,GOOD                ;IS RSCS2 CORRECT
016152  001401                                          BEQ     2$                      ;YES
016154  104000                                          HLT                             ;BAD=RSCS2 GOOD=CORRECT ANS
                                                                                        ;MXF SHOULD BE SET IN RSCS2
                                                                                        ;FOR A READ WAS ISSUED
                                                                                        ;WITH ERROR BITS SET IN RSER.
016156  022777  144270  162714  2$:             CMP     #144270,@RSCS1          ;IS RSCS1 CORRECT?
016164  001401                                          BEQ     3$                      ;YES
016166  104042                                          HLT     !DS!ER                  ;SC AND TRE SHOULD BE SET FOR
                                                                                        ;MXF SHOULD BE SET IN RSCS2
```

```
016170  022777  177777  162706  3$:     CMP     #-1,@RSWC       ;DID RSWC CHANGE?
016176  001401                          BEQ     .+4             ;NO
016200  104010                          HLT     !WC             ;WC SHOULD NOT HAVE CHANGED
                                                                ;FOR READ SHOULD HAVE NEVER
                                                                ;BEEN EXECUTED
016202  022777  027410  162676          CMP     #OUTBUF,@RSBA   ;DID RSBA MOVE
016210  001405                          BEQ     4$              ;NO
016212  017700  162670                  MOV     @RSBA,BAD       ;GET RSBA
016216  012701  027410                  MOV     #OUTBUF,GOOD    ;GET CORRECT ANS
016222  104000                          HLT                     ;RSBA=BAD GOOD=CORRECT ANS
                                                                ;RSBA MOVED WHEN THE READ
                                                                ;COMD WAS LOADED WITH ERROR
                                                                ;BITS SET IN RSER, READ COMD SHOULD
                                                                ;NEVER HAVE BEEN EXECUTED AND RSBA
                                                                ;SHOULD NOT HAVE MOVED.
016224  104414                  4$:     CLRDK                   ;CLEAR ALL REGISTERS
016226  017700  162650                  MOV     @RSCS2,BAD      ;GET RSCS2
016232  012701  000100                  MOV     #100,GOOD       ;SETUP CORRECT
016236  056701  162716                  BIS     UNNUM,GOOD      ;ANS
016242  020100                          CMP     GOOD,BAD        ;IS CS2 CORRECT?
016244  001401                          BEQ     .+4             ;YES
016246  104000                          HLT                     ;MXF SHOULD HAVE CLEARED IN
                                                                ;RSCS2 WITH THE "CLR" BIT BEING
                                                                ;SET IN RSCS2.
```

```
                                        ;*************************************************************
                                        ;TEST 57                    INVALID ADDRESS ERROR (IAE) TEST
                                        ;*************************************************************
016250  104400                          TST57:  SCOPE

                                        ;MODULE TESTED M7754, M7770
                                        ;FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK
                                        ;ADDRESS REGISTER (RSDA). THIS SHOULD CAUSE "IAE" TO SET IN
                                        ;THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO
                                        ;RSCS1 WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE
                                        ;DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE
                                        ;CONTROL REGISTER (RSCS1).
016252  042767  000040  162706                  BIC     #BIT5,ONCEE             ;CLEAR CLK CNT FLAG
016260  012702  004000                          MOV     #4000,R2                ;LOAD R2 WITH INVALID ADDR
016264  012767  016272  162516                  MOV     #4$,LAD                 ;LOOP TO HERE ON ERROR
016272  104416                          4$:     MRDMD                           ;PUT DRIVE IN MAINT MODE
016274  104420                                  MRCK                            ;CHECK MAINT REG
016276  022701                                  22701
016300  104424                                  MRINT                           ;INIT MAINT MODE (CLEAR MRSP)
016302  032767  000004  162656                  BIT     #BIT2,ONCEE             ;LOOPING ON ERRORS>
016310  001002                                  BNE     1$                      ;YES
016312  006102                                  ROL     R2                      ;GET INVALID ADDRESS
016314  103454                                  BCS     IADONE                  ;DONE FLOATING A ONE YET?
C16316  010277  162566                  1$:     MOV     R2,@RSDA                ;LOAD RSDA WITH INVALID ADDRESS
016322  012777  000071  162550                  MOV     #71,@RSCS1              ;DO A READ TO INVALID ADDR
016330  022777  002000  162556                  CMP     #2000,@RSER             ;IS RSER CORRECT?
016336  001404                                  BEQ     2$                      ;YES
016340  052767  000004  162620                  BIS     #BIT2,ONCEE             ;SET ERROR BIT
016346  104044                                  HLT     !DS!DA                  ;RSER SHOULD=2000 FOR
                                                                                ;A READ COMMAND WAS GIVEN
                                                                                ;TO AN ILLEGAL ADDRESS
016350  042767  000004  162610          2$:     BIC     #BIT2,ONCEE             ;CLEAR ERROR FLAG
016356  022777  150600  162526                  CMP     #150600,@RSDS           ;DID IAE SET?
016364  001404                                  BEQ     3$                      ;YES
016366  052767  000004  162572                  BIS     #BIT2,ONCEE             ;SET ERROR BIT
016374  104044                                  HLT     !DS!DA                  ;RSDS SHOULD=150600 FOR
                                                                                ;IAE SHOULD BE SET IN RSER
016376  042767  000004  162562          3$:     BIC     #BIT2,ONCEE             ;CLEAR ERROR FLAG
016404  032777  100000  162466                  BIT     #BIT15,@RSCS1           ;DID SC SET?
016412  001004                                  BNE     5$                      ;YES
016414  052767  000004  162544                  BIS     #BIT2,ONCEE             ;SET ERROR BIT
016422  104044                                  HLT     !DA!DS                  ;SC SHOULD BE SET IN RSCS1
                                                                                ;FOR IAE SHOULD BE SET IN RSER
016424  042767  000004  162534          5$:     BIC     #BIT2,ONCEE             ;CLEAR ERROR BIT
016432  104414                                  CLRDK                           ;CLEAR ALL ERRORS
016434  005777  162454                          TST     @RSER                   ;DID IAE CLEAR?
016440  001401                                  BEQ     .+4                     ;YES
016442  104040                                  HLT     !DS                     ;IAE DID NOT CLEAR
016444  000712                                  BR      4$                      ;CONTINUE
016446                                  IADONE:                                 ;DONE
```

```
                                        ;***************************************************************
                                        ;TEST 60                OPERATION INCOMPLETE ERROR TEST
                                        ;***************************************************************
016446  104400                          TST60:  SCOPE

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

016450  104414                          MROPI:  CLRDK                           ;CLEAR ALL DRIVE REGISTERS
016452  013777  027410  162426                  MOV     @#OUTBUF,@RSBA          ;SETUP RSBA
016460  012777  177777  162416                  MOV     #-1,@RSWC               ;SETUP RSWC

016466  104430                                  MRIND                           ;SEND INDEX PULSE TO MR REG
016470  104420                                  MRCK                            ;CHECK MAINT REG
016472  022701                                  22701                           ;TO EQUAL 22701
016474  104424                                  MRINT                           ;INIT MAINT MODE (CLEAR MRSP)

016476  012777  000071  162374                  MOV     #71,@RSCS1              ;LOAD A READ COMMAND

016504  104430                                  MRIND                           ;ISSUE THREE INDEX
016506  104430                                  MRIND                           ;PULSES TO
016510  104430                                  MRIND                           ;CAUSE OPI

                                        ;NOW CHECK FOR CORRECT ERRORS IN RSER AND RSDS
016512  017700  162376                          MOV     @RSER,BAD               ;GET RSER
016516  012701  020000                          MOV     #20000,GOOD             ;GET CORRECT ANS
016522  020100                                  CMP     GOOD,BAD                ;DID OPI SET IN RSER?
016524  001434                                  BEQ     1$                      ;YES
016526  104402  016532                          TYPE    ,.+2                    ;.ASCIZ <15><12>"OPI IN RSER SHOULD SET-3 INDEX PULSES W
016614  104000                                  HLT                             ;RSER=BAD GOOD=CORRECT ANS

016616  022777  150600  162266  1$:             CMP     #150600,@RSDS           ;DID CORRECT ERRORS SET?
016624  001401                                  BEQ     2$                      ;YES
016626  104040                                  HLT     !DS                     ;RSDS SHOULD=150600 BECAUSE
                                                                                ;OF OPI ERROR IN RSER
016630  022777  144270  162242  2$:             CMP     #144270,@RSCS1          ;DID SC AND TRE SET IN RSCS1?
016636  001401                                  BEQ     MROPIA                  ;YES
016640  104050                                  HLT     !DS!WC                  ;SC AND TRE SHOULD SET IN RSCS1
                                                                                ;BECAUSE OF ERROR IN RSER
016642  104414                          MROPIA: CLRDK                           ;CLEAR ALL ERRORS
016644  005777  162244                          TST     @RSER                   ;DID OPI CLEAR IN RSER
016650  001437                                  BEQ     1$                      ;YES
016652  104402  016656                          TYPE    ,.+2                    ;.ASCIZ <15><12>"OPI IN RSER DID NOT CLEAR BY SETTING CL
016746  104040                                  HLT     !DS                     ;RSER SHOULD=0
016750  022777  010600  162134  1$:             CMP     #10600,@RSDS            ;DID ERROR BITS CLEAR IN RSDS
                                                                                ;BY SETTING CLR BIT IN RSCS2
016756  001401                                  BEQ     .+4                     ;YES
016760  104040                                  HLT     !DS                     ;RSDS SHOULD=10600
```

```
                                      ;*******************************************************************
                                      ;TEST 61                 PARITY ERROR TEST
                                      ;*******************************************************************
 016762  104400                       TST61:  SCOPE

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

 016764  104414                       MRPAR:  CLRDK                    ;CLEAR ALL REGISTERS
 016766  042767  000040  162172               BIC     #BIT5,ONCEE      ;CLEAR CLK CNT FLAG
 016774  104430                               MRIND                    ;SEND INDEX PULSE TO MR REG
 016776  104420                               MRCK                     ;CHECK MAINT TO
 017000  022701                               22701                    ;EQUAL 22701
 017002  104424                               MRINT                    ;INIT MAINT MODE (CLEAR MRSP)
 017004  052777  000020  162070               BIS     #BIT4,@RSCS2     ;SET THE "PAT" BIT.
 017012  012777  000077  162070               MOV     #77,@RSDA        ;BY WRITING INTO THIS REGISTER,
                                                                       ;PAR SHOULD SET IN RSER
 017020  022777  000010  162066               CMP     #10,@RSER        ;DID PAR SET?
 017026  001401                               BEQ     .+4              ;YES
 017030  104040                               HLT     !DS              ;"PAR" IN RSER SHOULD BE SET FOR
                                                                       ;THE "PAT" BIT WAS SET IN RSCS2
                                                                       ;WHEN PROGRAM TRIED TO WRITE INTO RSDA
 017032  022777  104200  162040               CMP     #104200,@RSCS1   ;DID PAR CAUSE SC TO SET?
 017040  001401                               BEQ     .+4              ;YES
 017042  104044                               HLT     !DS!DA           ;SC SHOULD BE SET IN RSCS1 FOR
                                                                       ;PAR SHOULD BE SET IN RSER
 017044  022777  000077  162036               CMP     #77,@RSDA        ;DID RSDA GET LOADED?
 017052  001401                               BEQ     .+4              ;YES
 017054  104004                               HLT     !DA              ;RSDA SHOULD=77 FOR PAT
                                                                       ;BIT WAS SET WHEN PROGRAM
                                                                       ;TRIED TO WRITE INTO RSDA
 017056  104414                               CLRDK                    ;CLEAR ALL ERRORS
 017060  022777  004200  162012               CMP     #4200,@RSCS1     ;DID ERRORS CLEAR?
 017066  001401                               BEQ     .+4              ;YES
 017070  104044                               HLT     !DS!DA           ;SC DID NOT CLEAR BY USING
                                                                       ;THE "CLR" BIT IN RSCS2
 017072  005777  162016                       TST     @RSER            ;DID PAR CLEAR?
 017076  001401                               BEQ     .+4              ;YES
 017100  104044                               HLT     !DS!DA           ;PAR DID NOT CLEAR BY USING
                                                                       ;THE CLR BIT IN RSCS2
```

```
                                        ;*****************************************************************
                                        ;TEST 62                   MAINTENANCE MODE INTERRUPT TEST
                                        ;*****************************************************************
 017102  104400                         TST62:  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.

 017104  012767  000002  162032                 MOV     #2,FLAG2
 017112  104414                         MREX:   CLRDK                   ;CLEAR DRIVE REGISTERS
 017114  012737  000200  177776                 MOV     #200,@#PS       ;SETUP FOR INTERRUPT
 017122  012706  000500                         MOV     #500,SP
 017126  052767  000040  162032                 BIS     #BIT5,ONCEE     ;SET TYPE CLOCK COUNT FLAG
 017134  042767  000600  162024                 BIC     #600,ONCEE      ;CLEAR FLAG BITS
 017142  104430                                 MRIND                   ;SEND INDEX PULSE TO MR REG
 017144  104420                                 MRCK                    ;CHECK MR REG
 017146  022701                                 22701                   ;TO EQUAL 22701
 017150  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)
 017152  012702  026610                         MOV     #INBUF,R2       ;GET LOCATION OF OUTBUF
 017156  005022                                 CLR     (R2)+           ;CLEAR 1ST LOCATION
 017160  012722  177777                         MOV     #-1,(R2)+       ;2ND WORD OF ALL ONES
 017164  005003                                 CLR     R3              ;CLEAR WORK LOC TO GENERATE
 017166  000261                                 SEC                     ;A PATTERN OF FLOATING ONES
 017170  006103                         1$:     ROL     R3              ;GET PATTERN
 017172  103402                                 BCS     2$              ;DONE GET OUT
 017174  010322                                 MOV     R3,(R2)+        ;FILL BUFFER
 017176  000774                                 BR      1$              ;CONT
 017200  012703  000156                 2$:     MOV     #110.,R3        ;FILL REMAINING PORTION OF
 017204  012704  146314                         MOV     #146314,R4      ;BUFFER WITH A PATTERN OF 146314
 017210  010422                         3$:     MOV     R4,(R2)+        ;LOAD BUFFER
 017212  005303                                 DEC     R3              ;DONE YET?
 017214  001375                                 BNE     3$              ;NO

                                        ;SETUP CONTROLLER TO TRANSFER 256 WORDS OF DATA (2 SECTORS)
 017216  012777  020024  161704                 MOV     #INTMR,@RSVEC   ;SETUP INTERRUPT VECTOR
 017224  012777  000340  161700                 MOV     #340,@RSVCPS
 017232  012777  026610  161646                 MOV     #INBUF,@RSBA    ;LOAD BUS ADDR REG
 017240  012777  177400  161636                 MOV     #177400,@RSWC   ;LOAD WORD COUNT REG
 017246  012777  000161  161624                 MOV     #161,@RSCS1     ;LOAD WRITE COMMAND I/E
 017254  104446                                 GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
                                                                        ;TO CLEAR OUT COUNTERS AND REGISTERS
                                                                        ;THAT OTHERWISE COULD NOT BE CLEARED.
 017256  104220                                 HLT     !MR             ;COULD NOT SET SECTOR PULSE (0)
 017260  104450                                 SPASS                   ;CLOCK MR REG SP = 1
```

```
                                        ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
 017262  104430                                 MRIND
 017264  104420                                 MRCK                            ;CHECK MR REG TO EQUAL
 017266  020501                                 20501                           ;20501 FOR A
 017270  104000                                 HLT                             ;WRITE COMD

                                        ;STEP THRU RESYNC PERIOD

 017272  012767  001000  161702                 MOV     #512.,REPT
 017300  052767  000040  161660                 BIS     #BITS,ONCEE             ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
                                        MREX1:  MRCLK                           ;CLOCK MR REG
 017306  104422
 017310  104420                                 MRCK                            ;CHECK FOR
 017312  070501                                 70501                           ;CORRECT DATA
 017314  104000                                 HLT                             ;MR = BAD GOOD = CORRECT DATA
 017316  104422                                 MRCLK                           ;CLOCK MR REG
 017320  104420                                 MRCK                            ;CHECK FOR
 017322  020501                                 20501                           ;CORRECT DATA
 017324  104000                                 HLT                             ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
 017326  005367  161650                         DEC     REPT                    ;FINISH LOOPING
 017332  001365                                 BNE     MREX1                   ;THROUGH RESYNC PERIOD

                                        ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
                                        ;SP=0 EQUALS SECTOR PULSE
 017334  104422                                 MRCLK                           ;CLOCK MR REG
 017336  104420                                 MRCK                            ;MR SHOULD
 017340  070101                                 70101                           ;EQUAL 70101
 017342  104000                                 HLT                             ;MR=BAD GOOD=CORRECT ANS
 017344  104422                                 MRCLK                           ;CLOCK MR REG
 017346  104420                                 MRCK                            ;CHECK MR
 017350  020101                                 20101                           ;TO EQUAL 20101
 017352  104000                                 HLT                             ;MR=BAD GOOD=CORRECT ANS

                                        ;PERFORM 63 DOUBLE MAINT CLOCK OPERATIONS--WRITING PREAMBLE

 017354  012767  000077  161620                 MOV     #63.,REPT
 017362  104422                         MREX2:  MRCLK                           ;CLOCK MR REG
 017364  104420                                 MRCK                            ;CHECK MR REG
 017366  071501                                 71501                           ;TO EQUAL 71501
 017370  104000                                 HLT                             ;MR=BAD GOOD=CORRECT ANS
 017372  104422                                 MRCLK                           ;CLOCK MR REG
 017374  104420                                 MRCK                            ;CHECK MR REG
 017376  021501                                 21501                           ;TO EQUAL 21501
 017400  104000                                 HLT                             ;MR=BAD GOOD=CORRECT ANS
 017402  005367  161574                         DEC     REPT                    ;DONE YET
 017406  001365                                 BNE     MREX2                   ;NO LOOP
```

```
                                     ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN

017410  104422                                  MRCLK                   ;CLOCK MR REG
017412  104420                                  MRCK                    ;CHECK MR REG
017414  171501                                  171501                  ;TO EQUAL 171501
017416  104000                                  HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017420  104422                                  MRCLK                   ;CLOCK MR REG
017422  104420                                  MRCK                    ;MR REG SHOULD
017424  025501                                  25501                   ;EQUAL 25501
017426  104000                                  HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017430  104422                                  MRCLK
017432  104420                                  MRCK
017434  175501                                  175501
017436  104000                                  HLT
                                     ;PERFORM NEXT STEP 3 TIMES TO FINISH WRITTING PREAMBLE
017440  012767  000003  161534                  MOV     #3,REPT
017446  104422                       MREX3:     MRCLK                   ;CLOCK MR REG
017450  104420                                  MRCK                    ;CHECK MR REG
017452  025501                                  25501                   ;TO EQUAL 25501
017454  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017456  104422                                  MRCLK                   ;CLOCK MR REG
017460  104420                                  MRCK                    ;CHECK MR REG
017462  175501                                  175501                  ;TO EQUAL 175501
017464  104000                                  HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017466  005367  161510                          DEC     REPT            ;DONE YES?
017472  001365                                  BNE     MREX3           ;NO LOOP BACK

                                     ;MOVE DATA WORD INTO RS04 SHIFT REGISTER

017474  104422                                  MRCLK                   ;CLOCK MR REG
017476  104420                                  MRCK                    ;CHECK MR REG
017500  027501                                  27501                   ;TO EQUAL 27501
017502  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017504  104422                                  MRCLK                   ;CLOCK MR REG
017506  104420                                  MRCK                    ;MR REG SHOULD
017510  123501                                  123501                  ;EQUAL 123501
017512  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS

                                     ;ENCODE SYNC 1

017514  104422                                  MRCLK                   ;CLOCK MR REG
017516  104420                                  MRCK                    ;MR REG SHOULD NOW
017520  073501                                  73501                   ;EQUAL 73501
017522  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017524  012705  026610                          MOV     #INBUF,R5       ;GET STARTING ADDR FOR DATA BUFFER
017530  011504                                  MOV     (R5),R4         ;GET DATA
```

```
017532 012767  002167  161454            MOV     #1143.,WORK        ;DOING A 1 SECTOR TRANSFER 127 WORDS
                                                                    ;18 BITS PER WORD-CLOCK LOOPS
                                                                    ;TAKE CARE OF 2 BITS AT A TIME
                                                                    ;127 TIMES 9 EQUALS 1143 LOOPS
                                                                    ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
017540 042767  000200  161420            BIC     #BIT7,ONCEE        ;CLEAR LAST WORD FLAG
017546 052767  000100  161412            BIS     #BIT6,ONCEE        ;SET 1ST TRANSFER WORD FLAG
017554 104432                    1$:     XBIT                       ;GET 2 BITS OF DATA
017556 104434                            CLKD1                      ;SEND FIRST CLOCK PULSE
                                                                    ;AND CALCULATE MR REG
                                                                    ;FOR CORRECT DATA (MWDT+MWDB)
017560 104000                            HLT                        ;MR REG NOT CORRECT
017562 104436                            CLKD2                      ;SEND 2ND CLOCK PULSE TO
                                                                    ;COMPLETE TRANSFER OF 2 BITS
                                                                    ;CALCULATE CORRECT ANS FOR
                                                                    ;MR REG (MWDT+MWDB)
017564 104000                            HLT                        ;MR=BAD GOOD=CORRECT ANS
017566 032767  000200  161372            BIT     #BIT7,ONCEE        ;ON LAST WORD YET
017574 001015                            BNE     2$                 ;YES
017576 032767  000400  161362            BIT     #BIT8,ONCEE        ;ON CRC WORD YET?
017604 001043                            BNE     3$                 ;YES
017606 005367  161402                    DEC     WORK               ;DONE WITH 127 WORDS?
017612 001360                            BNE     1$                 ;NO

017614 052767  000200  161344            BIS     #BIT7,ONCEE        ;SET LAST WORD FLAG
017622 012767  000012  161364            MOV     #10.,WORK          ;SET UP TO TRANSFER LAST WORD
017630 005367  161360            2$:     DEC     WORK               ;DONE YET
017634 001347                            BNE     1$

017636 052767  000400  161322            BIS     #BIT8,ONCEE        ;SET TRANSFERRING CRC WORD
017644 042767  000200  161314            BIC     #BIT7,ONCEE        ;CLEAR LAST WORD FLAG

                                 ;GENERATE RMR ERROR BY ATTEMPTING TO WRITE RSER
                                 ;EXC SHOULD THEN BE ASSERTED

017652 012777  177777  161234            MOV     #-1,@RSER
017660 004767  003572                    JSR     PC,GENCRC          ;GENERATE CRC WORD
                                                                    ;AND LEAVE IN "WORK"
017664 012702  026610                    MOV     #INBUF,R2          ;GO TO END
017670 062702  000400                    ADD     #400,R2            ;OF DATA BUFFER
017674 016712  161314                    MOV     WORK,@R2           ;LOAD CRC WORD
017700 010205                            MOV     R2,R5              ;RESET POINTER FOR
017702 162705  000002                    SUB     #2,R5              ;R5 FOR CRC WD
017706 012767  000012  161300            MOV     #10.,WORK          ;SETUP TO XFER CRC
017714 005367  161274            3$:     DEC     WORK               ;DONE YET?
017720 001315                            BNE     1$                 ;NO
```

```
                                        ;EBL SHOULD NOW ASSERT AND CRC BE WRITTEN
017722  104422                                  MRCLK                   ;CLOCK MR REG TO STEP THROUGH DEAD BAND AREA
017724  104420                                  MRCK                    ;CHECK MR REG
017726  153501                                  153501                  ;TO EQUAL 103501
017730  104000                                  HLT                     ;MR REG=BAD GOOD=CORRECT ANS

                                        ;LOOP 6 TIMES
017732  012767  000006  161242                  MOV     #6,REPT
017740  104422                          4$:     MRCLK                   ;CLOCK MR REG
017742  104420                                  MRCK                    ;CHECK MR REG
017744  003501                                  3501                    ;TO EQUAL 53501
017746  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017750  104422                                  MRCLK                   ;CLOCK MR REG
017752  104420                                  MRCK                    ;CHECK MR REG
017754  153501                                  153501                  ;TO EQUAL 103501
017756  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
017760  005367  161216                          DEC     REPT            ;DONE LOOPING YET?
017764  001365                                  BNE     4$              ;NO

                                        ;FINISH UP
017766  104422                                  MRCLK                   ;CLOCK MR REG
017770  104420                                  MRCK                    ;CHECK MR REG
017772  003501                                  3501                    ;TO EQUAL 3501
017774  104000                                  HLT                     ;MR REG=BAD GOOD=CORRECT ANS
017776  104422                                  MRCLK                   ;CLOCK MR REG
020000  104420                                  MRCK                    ;CHECK MR REG
020002  151501                                  151501                  ;TO EQUAL 151501
020004  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS

                                        ;TRANSFER SHOULD NOW BE COMPLETE
020006  104422                                  MRCLK                   ;CLOCK MR REG
020010  104420                                  MRCK                    ;CHECK MRE REG
020012  002701                                  2701                    ;TO EQUAL 2701
020014  104000                                  HLT                     ;MR=BAD GOOD=CORRECT ANS
020016  000240                                  NOP                     ;STALL FOR TIME
020020  104050                                  HLT     !WC!DS          ;SHOULD NEVER GET HERE
020022  000424                                  BR      INTMR1          ;BECAUSE DRIVE SHOULD HAVE INTERRUPTED.
                                                                        ;CAUSING JUMP TO INTMR.
                                                                        ;CHECK FOR ASSERTION OF FTS ATTN L
                                        ;NOW TEST CONTROLLER

020024  022777  144260  161046          INTMR:  CMP     #144260,@RSCS1  ;IS CS1 CORRECT?
020032  001401                                  BEQ     .+4             ;YES
020034  104014                                  HLT     !DA!WC          ;YES
020036  022777  177610  161040          5$:     CMP     #177610,@RSWC   ;IS WC REG CORRECT?
020044  001401                                  BEQ     .+4             ;YES
020046  104010                                  HLT     !WC             ;WC SHOULD BE = TO 177610
020050  022777  000004  161036                  CMP     #4,@RSER        ;DID RMR SET IN RSER
020056  001401                                  BEQ     .+4             ;YES
020060  104050                                  HLT     !DS!WC          ;RSER SHOULD = 4
020062  022777  000001  161020                  CMP     #1,@RSDA        ;DOES RSDA=1
020070  001401                                  BEQ     .+4             ;YES
020072  104004                                  HLT     !DA             ;RSDA SHOULD=1
020074  000240                          INTMR1: NOP                     ;DONE
```

```
                                        ;*************************************************************
                                        ;TEST 63                  DISK ADDRESS OVERFLOW TEST
                                        ;*************************************************************
020076  104400                          TST63:  SCOPE

                                        ;MODULES TESTED: M7754, M7771, M7770
                                        ;SET UP TC 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.

020100  104414                          MRADE:  CLRUK                           ;CLEAR ALL REGISTERS
020102  012706  000500                          MOV     #500,SP                 ;SETUP STACK POINTER
020106  104430                                  MRIND                           ;SEND INDEX PULSE TO MR REG
020110  104420                                  MRCK                            ;CHECK MAINT REG
020112  022701                                  22701                           ;TO EQUAL 22701
020114  104424                                  MRINT                           ;INITIALIZE MAINT REG BY SENDING
                                                                                ;2 CLOCK PULSES (CLEAR MRSP)
020116  012777  007777  160764                  MCV     #7777,@RSDA             ;SETUP DISK ADDRESS
020124  012777  177400  160752                  MOV     #-400,@RSWC             ;SETUP FOR A 2 SECTOR TRANSFER
020132  012777  027410  160746                  MOV     #OUTBUF,@RSBA           ;GET OUTPUT BUFFER

                                        ;SETUP BUFFER WITH ALL ONES
020140  012705  027410                          MOV     #OUTBUF,R5              ;GET STARTING ADDRESS OF OUTBUF
020144  012767  000400  161030                  MOV     #400,REPT               ;LOAD 2 SECTORS
020152  012725  177777                  1$:     MOV     #-1,(R5)+               ;WITH WORDS
020156  005367  161020                          DEC     REPT                    ;OF ALL ONES
020162  001373                                  BNE     1$

020164  012777  000061  160706                  MOV     #61,@RSCS1              ;LOAD WRITE COMMAND
020172  104430                                  MRIND                           ;SET INDEX PULSE

                                        ;SUPPLY CLOCKS TO STEP THROUGH A TRACK

020174  012767  000003  161000                  MOV     #3,REPT
020202  012704  160000                  5$:     MOV     #57344.,R4              ;SETUP FOR FAST CLOCK PULSES 172032 CLOCKS
020206  012702  000011                          MOV     #11,R2                  ;(3 X 57344 = 172032)
020212  012703  000001                          MOV     #1,R3
020216  010277  160702                  2$:     MOV     R2,@RSMR
020222  010377  160676                          MOV     R3,@RSMR
020226  005304                                  DEC     R4
020230  001372                                  BNE     2$
020232  005367  160744                          DEC     REPT
020236  001361                                  BNE     5$

                                        ;CAUSE "LBT IN RSDS TO SET

020240  104422                                  MRCLK                           ;CLOCK AN 11 AND A 1 INTO RSMR
020242  104426                                  DSCK                            ;CHECK MR
020244  012400                                  12400                           ;TO EQUAL 12400
020246  104000                                  HLT                             ;LBT SHOULD BE SET IN RSDS
```

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

020250  104430                                   MRIND                       ;ASSERT MAINT INDEX PULSE
020252  005067  160716                           CLR     MCCNT               ;CLEAR THE CLOCK COUNTER
020256  104420                                   MRCK                        ;CHECK MR REG
020260  002501                                   2501                        ;TO EQUAL 2501. SHOULD STILL BE WRITING
020262  104000                                   HLT

                                          ;SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS04 RESYNC PERIOD
020264  012767  001000  160710                   MOV     #512.,REPT          ;CLOCK COUNT TO STEP THRU RESYNC
020272  104422                            4$:    MRCLK                       ;2ND REVOLUTION
020274  104420                                   MRCK                        ;CHECK MR
020276  052501                                   52501                       ;TO EQUAL 52501
020300  104000                                   HLT                         ;MR=BAD GOOD=CORRECT ANS
020302  104422                                   MRCLK                       ;CLOCK MR REG
020304  104420                                   MRCK                        ;CHECK MR
020306  002501                                   2501                        ;REG TO
020310  104000                                   HLT                         ;EQUAL 2501
020312  005367  160664                           DEC     REPT
020316  001365                                   BNE     4$                  ;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

020320  104422                            AOECK: MRCLK
020322  104422                                   MRCLK                       ;CAUSE SECTOR PULSE AND AOE ERROR
020324  104420                                   MRCK                        ;CHECK FOR SECTOR PULSE
020326  022301                                   22301                       ;IN RSMR
020330  104000                                   HLT                         ;MR=BAD GOOD=CORRECT ANS
020332  022777  001000  160554                   CMP     #1000,@RSER         ;DID AOE SET IN RSER?
020340  001401                                   BEQ     1$                  ;AOE SHOULD BE SET IN RSER
020342  104040                                   HLT     !DS                 ;RSER SHOULD EQUAL 1000
020344  022777  152600  160540            1$:    CMP     #152600,@RSDS       ;IS RSDS CORRECT
020352  001401                                   BEQ     2$                  ;YES
020354  104040                                   HLT     !DS                 ;ERR & ATA SHOULD BE SET IN RSDS
                                                                             ;BECAUSE OF AOE ERROR IN RSER
020356  104414                            2$:    CLRDK                       ;CLEAR ERROR
020360  005777  160530                           TST     @RSER               ;DID ERROR CLEAR?
020364  001401                                   BEQ     3$                  ;YES
020366  104040                                   HLT     !DS                 ;AOE DID NOT CLEAR BY SETTING CLR IN RSCS2
020370  022777  010600  160514            3$:    CMP     #10600,@RSDS        ;DID ERRORS CLEAR
020376  001401                                   BEQ     .+4                 ;YES
020400  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 64                  MAINTENANCE MODE VERIFY TEST
                                          ;*******************************************************************
020402  104400                            TST64:  SCOPE

                                          ;MODULE TESTED G182

020404  032767  004000  157156            MRVR:   BIT     #BIT11,SWR        ;DO THIS TEST?
020412  001002                                    BNE     3$                ;YES
020414  000137  021130                            JMP     @#INFTST          ;NO
020420  005067  160520                    3$:     CLR     FLAG2             ;SET VERIFY TEST FLAG
020424  104414                                    CLRDK                     ;CLEAR ALL DRIVES
020426  012767  177777  160574                    MOV     #177777,WORK5     ;STALL TO
020434  005367  160570                    4$:     DEC     WORK5             ;RESYNC DRIVE
020440  001375                                    BNE     4$                ;TIMING LOGIC
020442  042767  000040  160516                    BIC     #BIT5,ONCEE       ;CLEAR CLK CNT
020450  012777  160000  160426                    MOV     #-20000,@RSWC     ;WRITE ONE TRACK - 8K WDS
020456  012767  177777  006124                    MOV     #177777,INBUF     ;WRITE A PATTERN 12525
020464  052777  000010  160410                    BIS     #BIT3,@RSCS2      ;SET BAI BIT
020472  012777  026610  160406                    MOV     #INBUF,@RSBA      ;SET DATA WD
020500  012767  177777  160474                    MOV     #177777,REPT      ;SETUP WAIT LOOP
020506  012777  000061  160364                    MOV     #61,@RSCS1        ;GO WRITE
020514  105777  160360                    1$:     TSTB    @RSCS1            ;DONE YET?
020520  100404                                    BMI     2$                ;YES
020522  005367  160454                            DEC     REPT              ;DECREMENT COUNTER WAITING
020526  001372                                    BNE     1$                ;FOR READY
020530  104000                                    HLT                       ;READY NEVER CAME UP
020532  005777  160342                    2$:     TST     @RSCS1            ;ANY ERRORS?
020536  100002                                    BPL     MRVR1             ;NO
020540  104050                                    HLT     !DS!WC            ;STOP HERE TILL THIS PROBLEM IS FIXED TRY DZRSB DIAG
020542  000433                                    BR      TBDIA             ;TYPE MESSAGE
```

```
020544  104414                          MRVR1:  CLRDK                           ;CLEAR ALL REGISTERS
020546  012777  160000  160330                  MOV     #-20000,@RSWC           ;SETUP WC
020554  052777  000010  160320                  BIS     #BIT3,@RSCS2            ;SET BAI
020562  012777  026610  160316                  MOV     #INBUF,@RSBA            ;SETUP RSBA
020570  012767  177777  160404                  MOV     #177777,REPT            ;SETUP WAIT LOOP
020576  012777  000051  160274                  MOV     #51,@RSCS1              ;DO A WRITE CHECK TO VERIFY DISK
020604  105777  160270                  1$:     TSTB    @RSCS1                  ;TEST
020610  100404                                  BMI     2$                      ;FOR READY TO COME BACK
020612  005367  160364                          DEC     REPT                    ;WAIT
020616  001372                                  BNE     1$                      ;
020620  104000                                  HLT                             ;READY NEVER CAME BACK
020622  005777  160252                  2$:     TST     @RSCS1                  ;ANY ERRORS?
020626  100032                                  BPL     MRVRR                   ;NO
020630  104050                                  HLT     !DS!WC                  ;STOP HERE WC FAILED
                                                                                ;GO TO DZRSB DIAG
                                                                                ;BEFORE TRYING TO DEBUG THLIS TEST
020632                                  TBDIA:
020632  104402  020636                          TYPE    ,.+2                    ;.ASCIZ <15><12>"FAILED VERIFY TEST --- RUN DZRSB DIAGNO
020714  000137  011456                  MRVRR:  JMP     @#MRWRT                 ;GO WRITE IN MAINTENANCE MODE
                                        ;NOW CHECK TO SEE IF DRIVE WAS WRITTEN ON IN MAINTENANCE MODE

020720  104414                          MRVR2:  CLRDK                           ;CLEAR ALL REGISTERS
020722  012767  177777  160264                  MOV     #177777,WORK            ;STALL - TO RESPONSE
020730  005367  160260                  3$:     DEC     WORK                    ;INDEX PULSE
020734  001375                                  BNE     3$                      ;ON DRIVE
020736  012777  160000  160140                  MOV     #-20000,@RSWC           ;SETUP WC FOR 1 TRACK
020744  052777  000010  160130                  BIS     #BAI,@RSCS2             ;SET BAI
020752  012777  026610  160126                  MOV     #INBUF,@RSBA            ;SETUP RSBA
020760  012767  177777  005622                  MOV     #177777,INBUF           ;SETUP FOR COMPARE
020766  012777  000051  160104                  MOV     #51,@RSCS1              ;DO A WRITE CHECK
020774  105777  160100                  1$:     TSTB    @RSCS1                  ;TEST FOR
021000  100375                                  BPL     1$                      ;READY TO COME BACK
021002  032777  040000  160072                  BIT     #WCE,@RSCS2             ;DID WCE SET?
021010  001442                                  BEQ     2$                      ;NO
021012  104402  021016                          TYPE    ,.+2                    ;.ASCIZ <15><12> "WRITE AMPLIFIER DID NOT GET DISABLED B
021112  104040                                  HLT     !DS
021114  000404                                  BR      4$                      ;GET OUT
021116  005777  157756                  2$:     TST     @RSCS1                  ;ANY ERRORS?
021122  100001                                  BPL     4$                      ;NO
021124  104040                                  HLT     !DS                     ;SHOULD NOT BE ANY ERRORS
                                                                                ;TRY THE DZRSB DIAGNOSTIC
021126  000240                          4$:     NOP

021130  000137  002116                  INFTST: JMP     @#TRYNX                 ;GET NEXT DRIVE
```

```
                                          .SBTTL        $DONE - BELL AND SCOPE ROUTINE

021134  104400                            $DONE:  SCOPE                         ;TERMINATIONG SCOPE FOR LOOPING
021136  062767  000001  157642                    ADD     #1,$PCNT+2            ;ADD 1 TO THE PASS COUNT
021144  005567  157634                            ADC     $PCNT                 ;MAKE IT DOUBLE PREC.
021150  032737  002000  177570                    BIT     #SW10,@#SWR           ;RING THE BELL?
021156  001004                                    BNE     4$                    ;NO!
021160  104402  021164                            TYPE    ,.+2                  ;.ASCIZ <BELL><177>
021170  013700  000042                    4$:     MOV     @#42,R0               ;GET MONITOR ADDRESS
021174  001405                                    BEQ     $END1                 ;IF NONE
021176  000005                                    RESET
021200  004710                            $ENDAD: JSR     7,(0)                 ;GO TO MONITOR
021202  000240  000240  000240                    240,240,240                   ;SAVE ROOM FOR ACT11
021210  000167  000002                    $END1:  JMP     MULSYS                ;RETURN

021214  000000                            .TBIT:  0                             ;T BIT FLAG

                                          ;MULTI DRIVE SYSTEM?

021216                                    MULSYS:
021216  104402  021222                            TYPE    ,.+2                  ;.ASCIZ <15><12>"END OF PASS"
021240  005067  157544                            CLR     LAD
021244  005067  157530                            CLR     ICNT
021250  032767  000020  157710                    BIT     #BIT4,ONCEE           ;MULTI DRVIE?
021256  001002                                    BNE     1$                    ;NO
021260  000137  001672                            JMP     @#MULTII              ;YES
021264  000137  002322                    1$:     JMP     @#NOWGO               ;TEST ONLY ONE DRIVE

                                          ;ERROR TYPEOUT ROUTINE FOR NO-OP TEST

021270  032767  000004  157670            NOPERR: BIT     #BIT2,ONCEE           ;WERE WE HERE BEFORE?
021276  001031                                    BNE     1$                    ;YES
021300  052767  000004  157660                    BIS     #BIT2,ONCEE           ;SET BEEN HERE BEFORE FLAG
021306  104402  021312                            TYPE    ,.+2                  ;.ASCIZ <15><12>"ERROR CAUSED BY NO-OP FUNCTION "
021354  016746  157634                            MOV     WOR' -(6)             ;PUT WORK ON STACK
021360  104406                                    TYPES                         ;TYPE STACK IN OCTAL - SUPRESS
021362  000207                            1$:     RTS     PC
```

```
021364  104402  021400           CHG:    TYPE    ,REGCHG         ;TYPE MESSAGE
021370  016746  157620                   MOV     WORK,-(6)       ;PUT WORK ON STACK
021374  104406                           TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
021376  000207                           RTS     PC

021400  044103  047101  042507   REGCHG: .ASCIZ  "CHANGED WITH NO-OP FUNCTION "
021406  020104  044527  044124
021414  047040  026517  050117
021422  043040  047125  052103
021430  047511  020116     000

021435     015  051012  051115   TRMR:   .ASCIZ  <15><12>"RMR  DID NOT SET BY WRITING INTO "
021442  020040  044504  020104
021450  047516  020124  042523
021456  020124  054502  053440
021464  044522  044524  043516
021472  044440  052116  020117
021500     000
        021502                           .EVEN

021502  104422                   .MRINT: MRCLK                   ;CLOCK THE MAINT REG WITH AN 11 AND A 1
021504  104422                           MRCLK                   ;SAME
021506  000002                           RTI                     ;RETURN

021510  012777  000011  157406   .MRCLK: MOV     #11,@RSMR       ;CLOCK THE
021516  012777  000001  157400           MOV     #1,@RSMR        ;MAINT REG
021524  062767  000001  157444           ADD     #1,MCCNT+2      ;ADD 1 TO CLOCK COUNT
021532  005567  157436                   ADC     MCCNT           ;MAKE DOUBLE PRECISION
021536  000002                           RTI

021540  017700  157360           .MRCK:  MOV     @RSMR,BAD       ;GET THE CONTENTS OF RSMR
021544  017601  000000                   MOV     @(SP),GOOD      ;GET THE CORRECT ANSWER
021550  062716  000002                   ADD     #2,(SP)         ;UPDATE THE RETURN ADDRESS FOR AN ERROR
021554  020100                           CMP     GOOD,BAD        ;IS THE MR REG CORRECT?
021556  001002                           BNE     1$              ;NO EXIT
021560  062716  000002                   ADD     #2,(SP)         ;UPDATE RETURN ADDRESS TO SKIP THE HLT FOR CORRECT ANS
021564  000002                   1$:     RTI                     ;RETURN

                                 ;SEND INDEX PULSE TO THE MAINTENANCE REGISTER
021566  012777  000021  157330   .MRIND: MOV     #21,@RSMR       ;SEND INDEX
021574  012777  000001  157322           MOV     #1,@RSMR        ;PULSE TO MR REG
021602  000002                           RTI
021604  017700  157302           .DSCK:  MOV     @RSDS,BAD       ;GET THE CONTENTS OF RSDS
021610  017601  000000                   MOV     @(SP),GOOD      ;GET THE CORRECT ANS
021614  062716  000002                   ADD     #2,(SP)         ;UPDATE THE RETURN ADDR FOR AN ERROR
021620  020100                           CMP     GOOD,BAD        ;IS RSDS CORRECT
021622  001002                           BNE     1$              ;NO EXIT
021624  062716  000002                   ADD     #2,(SP)         ;UPDATE RETURN ADDR TO SKIP THE HLT FOR CORRRECT ANS
021630  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

021632  032767  000100  157326  .XBIT:  BIT     #BIT6,ONCEE     ;1ST 2 BITS OF 1ST WORD?
021640  001427                          BEQ     2$              ;NO
021642  012767  000001  157276          MOV     #1,LSTEV        ;SET LAST EVEN BIT TRANSFERRED TO A 1
021650  012767  000001  157272          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
021656  042767  000100  157302          BIC     #BIT6,ONCEE     ;CLEAR 1ST WORD TRANSFER FLAG
021664  005067  157316          4$:     CLR     CLKCNT          ;CLEAR CLOCK COUNTER AT START OF EACH WORD
021670  032767  000400  157270          BIT     #BIT8,ONCEE     ;CRC WORD BEING WRITTEN?
021676  001042                          BNE     1$              ;YES
021700  005067  157250                  CLR     NOWOD           ;NO, LOAD EVEN
021704  005067  157242                  CLR     NOWEV           ;AND ODD WITH 0 FOR BITS 16 & 17 IN RS04 DATA WORD.
021710  012767  000010  157306  6$:     MOV     #8.,WORK3       ;8 LOOPS FOR REMAINING 16 BITS OF WORD
021716  000002                          RTI
021720  016767  157230  157222  2$:     MOV     NOWOD,LSTOD
021726  016767  157220  157212          MOV     NOWEV,LSTEV     ;SAVE LAST 2 BITS TRANSFERRED
021734  005767  157264                  TST     WORK3           ;DONE WITH WORD YET?
021740  001004                          BNE     3$              ;NO
021742  062705  000002                  ADD     #2,R5           ;UPDATE BUFFER WD
021746  011504                          MOV     (R5),R4         ;GET DATA WD
021750  000745                          BR      4$              ;GET BITS 16 & 17
021752  005067  157176          3$:     CLR     NOWOD           ;CLEAR PRESENT ODD BIT
021756  006104                          ROL     R4              ;GET NEXT ODD DATA BIT
021760  006167  157170                  ROL     NOWOD           ;SAVE IT IN ODD BIT
021764  005067  157162                  CLR     NOWEV           ;CLEAR PRESENT EVEN BIT
021770  006104                          ROL     R4              ;GET NEXT EVEN BIT
021772  006167  157154                  ROL     NOWEV           ;SAVE IT IN EVEN BIT
021776  005367  157222                  DEC     WORK3           ;KEEP COUNT OF BITS IN THE WORD
022002  000002                          RTI                     ;RETURN

                                ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS. 0 & 1 ARE ALWAYS 0
022004  005067  157144          1$:     CLR     NOWOD           ;GET BITS 17
022010  006104                          ROL     R4              ;AND 16
022012  006167  157136                  ROL     NOWOD           ;FOR CRC WORD
022016  005067  157130                  CLR     NOWEV
022022  006104                          ROL     R4
022024  006167  157122                  ROL     NOWEV
022030  000727                          BR      6$              ;CONTINUE
```

```
                                        ;CLOCK ROUTINE (1ST OF TWO) WHICH IS USED TO CLOCK TWO BITS OF
                                        ;DATA TO THE DRIVE AT A TIME.  THIS ROUTINE ALSO CHECKS THE PREVIOUS
                                        ;BITS THAT HAVE BEEN TRANSFERRED AND CALCULATES WHICH STATE
                                        ;THE MWDT BIT (BIT 14 IN THE MR REG) AND MWDB BIT (BIT 12 IN THE MR REG) SHOULD BE IN


022032  104422                          .CLKD1: MRCLK                   ;CLOCK MR REG WITH AN 11 AND A 1
022034  005003                                  CLR     R3              ;CLEAR WORK LOCATION
022036  005767  157112                          TST     NOWOD           ;TEST ODD BIT NOW BEING SENT FOR A 1 OR A 0
022042  001005                                  BNE     TSTEVB          ;NOW TEST EVEN DATA BIT ON 1ST CLOCK
                                                                        ;NOW BIT IS A 1 MWDB IS 0
022044  005767  157100                  1$:     TST     LSTOD           ;TEST THE LAST ODD DATA BIT THAT WAS SENT
022050  001002                                  BNE     TSTEVB          ;LAST ODD DATA BIT WAS A 1
                                                                        ;MWDB IS A 0

022052  052703  010000                  2$:     BIS     #BIT12,R3       ;SET MWDB FOR LATER COMPARE WITH MR REG

                                        ;NOW TEST FOR EVEN BITS BEING SENT

022056  005767  157070                  TSTEVB: TST     NOWEV           ;TEST EVEN BIT NOW BEING TRANSFERRED
                                                                        ;FOR EITHER A 1 OR A 0
022062  001005                                  BNE     1$              ;NOW BIT IS A 1
022064  005767  157056                          TST     LSTEV           ;WAS LAST EVEN DATA BIT A 0?
022070  001002                                  BNE     1$              ;NO LAST EVEN DATA BIT WAS A 1
022072  052703  040000                          BIS     #BIT14,R3       ;MWDT SHOULD BE SET
022076  012701  123501                  1$:     MOV     #123501,GOOD    ;GET CORRECT ANS
022102  050301                                  BIS     R3,GOOD         ;FOR MR REG
022104  004767  001200                          JSR     PC,MRCAL        ;DETERMINE STATE OF SB & LSR BITS
022110  017700  157010                          MOV     @RSMR,BAD       ;GET CONTENTS OF MR REG
022114  020100                                  CMP     GOOD,BAD        ;IS MR REG CORRECT?
022116  001002                                  BNE     2$              ;NO TYPE OUT MR REG
022120  062716  000002                          ADD     #2,(SP)         ;UPDATE RETURN ADDR FOR CORRECT ANS
022124  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

022126  104422                          .CLKD2: MRCLK                           ;CLOCK MR REG
022130  005767  157020                          TST     NOWOD                   ;IS THE PRESENT DATA BIT A 1?
022134  001403                                  BEQ     1$                      ;NO IT IS A 0
022136  052703  010000                          BIS     #BIT12,R3               ;SET MWDB FOR BIT BEING SENT IS A 1
022142  000402                                  BR      2$
022144  042703  010000                  1$:     BIC     #BIT12,R3               ;CLEAR MWDB FOR PRESENT BIT IS A 0
022150  005767  156776                  2$:     TST     NOWEV                   ;IS PRESENT EVEN BIT A 1
022154  001403                                  BEQ     3$                      ;NO IT IS A 0
022156  052703  040000                          BIS     #BIT14,R3               ;IT IS A 1 SET MWDT
022162  000402                                  BR      4$
022164  042703  040000                  3$:     BIC     #BIT14,R3               ;PRESENT BIT IS A 0 CLEAR MWDT
022170  012701  023501                  4$:     MOV     #23501,GOOD             ;GET CORRECT ANS
022174  050301                                  BIS     R3,GOOD                 ;FOR MR REG
022176  004767  001106                          JSR     PC,MRCAL                ;DETERMINE STATE OF SB & LSR BITS
022202  017700  156716                          MOV     @RSMR,BAD               ;GET CONTENTS OF MR REG
022206  020100                                  CMP     GOOD,BAD                ;IS MR REG CORRECT?
022210  001002                                  BNE     5$                      ;NO TYPEOUT ERROR
022212  062716  000002                          ADD     #2,(SP)                 ;UPDATE RETURN ADDR FOR CORRECT ANS
022216  000002                          5$:     RTI                             ;RETURN

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

022220  012767  022330  156766          CRCTYP: MOV     #CRCTAB,WORK            ;GET STARTING LOC OF IC TABLE
022226  012767  000001  156764                  MOV     #1,WORK1                ;SETUP TO TEST FIRST CHIP
022234  036767  156760  156730          1$:     BIT     WORK1,SAVEE             ;WAS IT THIS BIT?
022242  001006                                  BNE     2$                      ;YES TYPE IT
022244  062767  000006  156742                  ADD     #6,WORK                 ;NO INDEX TABLE POINTER
022252  006167  156742                          ROL     WORK1                   ;SETUP TO TEST NEXT CHIP
022256  000766                                  BR      1$                      ;NOW TES IT
022260  004777  156730                  2$:     JSR     PC,@WORK                ;TYPE OUT CHIP
022264  104402  022270                          TYPE    ,.+2                    ;.ASCIZ " IN THE CRC REG SHOULD BE SET"
022326  000207                                  RTS     PC
```

```
                                         ;TABLE FOR CRC TEST 2 TYPEOUT ROUTINE

022330  104402  022470          CRCTAB: TYPE    ,E302
022334  000207                          RTS     PC
022336  104402  022476                  TYPE    ,E305
022342  000207                          RTS     PC
022344  104402  022504                  TYPE    ,E307
022350  000207                          RTS     PC
022352  104402  022512                  TYPE    ,E3010
022356  000207                          RTS     PC
022360  104402  022521                  TYPE    ,E3012
022364  000207                          RTS     PC
022366  104402  022530                  TYPE    ,E3015
022372  000207                          RTS     PC
022374  104402  022537                  TYPE    ,E242
022400  000207                          RTS     PC
022402  104402  022545                  TYPE    ,E245
022406  000207                          RTS     PC
022410  104402  022553                  TYPE    ,E247
022414  000207                          RTS     PC
022416  104402  022561                  TYPE    ,E2410
022422  000207                          RTS     PC
022424  104402  022570                  TYPE    ,E2412
022430  000207                          RTS     PC
022432  104402  022577                  TYPE    ,E2415
022436  000207                          RTS     PC
022440  104402  022606                  TYPE    ,E192
022444  000207                          RTS     PC
022446  104402  022614                  TYPE    ,E197
022452  000207                          RTS     PC
022454  104402  022622                  TYPE    ,E1910
022460  000207                          RTS     PC
022462  104402  022631                  TYPE    ,E1915
022466  000207                          RTS     PC
```

```
 4377  022470  031505  026460  000062  E302:    .ASCIZ  "E30-2"
 4378  022476  031505  026460  000065  E305:    .ASCIZ  "E30-5"
 4379  022504  031505  026460  000067  E307:    .ASCIZ  "E30-7"
 4380  022512  031505  026460  030061  E3010:   .ASCIZ  "E30-10"
 4381  022520     000
 4382  022521     105  030063  030455  E3012:   .ASCIZ  "E30-12"
 4383  022526  000062
 4384  022530  031505  026460  032461  E3015:   .ASCIZ  "E30-15"
 4385  022536     000
 4386  022537     105  032062  031055  E242:    .ASCIZ  "E24-2"
 4387  022544     000
 4388  022545     105  032062  032455  E245:    .ASCIZ  "E24-5"
 4389  022552     000
 4390  022553     105  032062  033455  E247:    .ASCIZ  "E24-7"
 4391  022560     000
 4392  022561     105  032062  030455  E2410:   .ASCIZ  "E24-10"
 4393  022566  000060
 4394  022570  031105  026464  031061  E2412:   .ASCIZ  "E24-12"
 4395  022576     000
 4396  022577     105  032062  030455  E2415:   .ASCIZ  "E24-15"
 4397  022604  000065
 4398  022606  030505  026471  000062  E192:    .ASCIZ  "E19-2"
 4399  022614  030505  026471  000067  E197:    .ASCIZ  "E19-7"
 4400  022622  030505  026471  030061  E1910:   .ASCIZ  "E19-10"
 4401  022630     000
 4402  022631     105  034461  030455  E1915:   .ASCIZ  "E19-15"
 4403  022636  000065
```

```
                                        ;GET TWO BITS OF DATA FROM INBUF
                                        ;FOR READING FROM DRIVE TO DETERMINE THE
                                        ;STATE OF MROT AND MROB IN THE MR REG.

022640  005767  156360                  .RBIT:  TST     WORK3           ;STARTING NEW WD?
022644  001031                                  BNE     3$              ;NO
022646  062705  000002                          ADD     #2,R5           ;UPDATE BUFFER WD
022652  011504                                  MOV     (R5),R4         ;GET DATA WD
022654  005067  156326                          CLR     CLKCNT          ;CLEAR CLOCK COUNTER AT START OF EACH WD
022660  032767  000400  156300                  BIT     #BIT8,ONCEE     ;ON CRC WD?
022666  001035                                  BNE     1$              ;YES
022670  032767  000040  156246                  BIT     #BIT5,FLAG2     ;IN CRC TEST ????
022676  001404                                  BEQ     7$              ;NO
022700  012767  000010  156316                  MOV     #8.,WORK3
022706  000410                                  BR      3$
022710  005067  156240                  7$:     CLR     NOWOD           ;LOAD EVEN & ODD BITS
022714  005067  156232                          CLR     NOWEV           ;WITH 0 FOR BITS 16 & 17 IN RS04 DATA WORD
022720  012767  000010  156276          6$:     MOV     #8.,WORK3       ;8 LOOPS FOR REMAINING 16 BITS OF WORD
022726  000002                                  RTI
022730  005067  156220                  3$:     CLR     NOWOD           ;CLEAR PRESENT ODD BIT
022734  006104                                  ROL     R4              ;GET NEXT ODD DATA BIT
022736  006167  156212                          ROL     NOWOD           ;SAVE IT IN ODD BIT
022742  005067  156204                          CLR     NOWEV           ;CLEAR PRESENT EVEN BIT
022746  006104                                  ROL     R4              ;GET NEXT EVEN BIT
022750  006167  156176                          ROL     NOWEV           ;SAVE IT IN EVEN BIT
022754  005367  156244                          DEC     WORK3           ;KEEP COUNT OF BITS IN THE WORD
022760  000002                                  RTI                     ;RETURN
                                        ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS, 0 & 1 ARE ALWAYS 0
022762  005067  156166                  1$:     CLR     NOWOD           ;GET BITS 17
022766  006104                                  ROL     R4              ;AND 16
022770  006167  156160                          ROL     NOWOD           ;FOR CRC WORD
022774  005067  156152                          CLR     NOWEV
023000  006104                                  ROL     R4
023002  006167  156144                          ROL     NOWEV
023006  000744                                  BR      6$              ;CONTINUE
```

```
4439  023010  004767  000236          .CLKR1: JSR     PC,CALRTB       ;CALCULATE TOP AND BOTTOM BITS FOR MR REG
4440  023014  012703  000011                  MOV     #11,R3          ;SETUP CLOCK BITS
4441  023020  056703  156170          CLOOK:  BIS     WORK,R3         ;SET TOP & BOTTOM BITS
4442  023024  010377  156074                  MOV     R3,@RSMR        ;SEND
4443  023030  042703  000010                  BIC     #BIT3,R3        ;CLOCK
4444  023034  010377  156064                  MOV     R3,@RSMR        ;PULSE
4445  023040  062767  000001  156130          ADD     #1,MCCNT+2      ;INCREMENT
4446  023046  005567  156122                  ADC     MCCNT           ;CLOCK COUNT
4447  023052  012701  023601                  MOV     #23601,GOOD     ;CALCULATE CORRECT ANS FOR MR REG
4448  023056  032767  000004  156060          BIT     #BIT2,FLAG2     ;WRITE CK TEST?
4449  023064  001402                          BEQ     7$              ;NO
4450  023066  052701  000100                  BIS     #BIT6,GOOD      ;YES SET RD IN MR REG
4451  023072  050301                  7$:     BIS     R3,GOOD
4452  023074  042701  000010                  BIC     #BIT3,GOOD      ;CLEAR MCLK
4453  023100  032767  000400  156060          BIT     #BIT8,ONCEE     ;ON CRC WD?
4454  023106  001406                          BEQ     5$              ;NO
4455  023110  022767  000011  156064          CMP     #11,REPT        ;SHOULD CRCW BE SET?
4456  023116  001402                          BEQ     5$              ;YES
4457  023120  042701  020000                  BIC     #20000,GOOD     ;CLEAR CRCW
4458  023124  032767  000001  156012  5$:     BIT     #BIT0,FLAG2     ;SHOULD SDCLK BE SET?
4459  023132  001004                          BNE     1$              ;YES
4460  023134  052767  000001  156002          BIS     #BIT0,FLAG2     ;NO
4461  023142  000405                          BR      2$              ;CONTINUE
4462  023144  052701  100000          1$:     BIS     #BIT15,GOOD     ;SET IT
4463  023150  042767  000001  155766          BIC     #BIT0,FLAG2     ;CLEAR FLAG FOR SDCLK FOR NEXT CLOCK PULSE
4464  023156  005367  156022          2$:     DEC     REPT1           ;SHOULD SB SET?
4465  023162  001017                          BNE     6$              ;NO
4466  023164  012767  000022  156012          MOV     #18.,REPT1      ;RESET SB COUNTER
4467  023172  052701  004000                  BIS     #BIT11,GOOD     ;SET SB
4468  023176  032767  000400  155762  3$:     BIT     #BIT8,ONCEE     ;ON CRC WD?
4469  023204  001406                          BEQ     6$              ;NO
4470  023206  022767  000022  155770          CMP     #22,REPT1       ;SHOULD SB AND CRCW BE SET ?
4471  023214  001002                          BNE     6$              ;NO
4472  023216  052701  020000                  BIS     #20000,GOOD     ;SET SB AND CRCW
4473  023222  017700  155676          6$:     MOV     @RSMR,BAD       ;GET MR REG
4474  023226  020100                          CMP     GOOD,BAD        ;IS RSMR CORRECT?
4475  023230  001002                          BNE     4$              ;NO
4476  023232  062716  000002                  ADD     #2,(SP)         ;YES
4477  023236  000002                  4$:     RTI                     ;RETURN
4478
4479  023240  004767  000006          .CLKR2: JSR     PC,CALRTB
4480  023244  012703  050011                  MOV     #50011,R3
4481  023250  000663                          BR      CLODK
```

```
                                        ;CALCULATE THE STATE OF MRDT AND MRDB FROM CURRENT INPUT BITS
                                        ;LOCATION WORK CONTAINS CORRECT DATA FOR MRDT AND MRDB
023252  005067  155736                  CALRTB: CLR     WORK            ;CLEAR WORK LOCATION
023256  005767  155672                          TST     NOWOD           ;IS CURRENT ODD BIT A 0?
023262  001403                                  BEQ     1$              ;YES
023264  052767  000004  155722                  BIS     #BIT2,WORK      ;NO SET MRDB
023272  005767  155654                  1$:     TST     NOWEV           ;IS CURRENT EVEN BIT A 0?
023276  001403                                  BEQ     2$              ;YES
023300  052767  000040  155706                  BIS     #BIT5,WORK      ;NO, SET MRDT
023306  000207                          2$:     RTS     PC              ;RETURN

                                        ;CALCULATE MR REG TO DETERMINE THE STATE OF THE CRC-SB AND LSR BITS
                                        ;ON THE DIFFERENT CLOCKS ON THE DIFFERENT WORDS THROUGHOUT THE SECTOR

023310  005267  155672                  MRCAL:  INC     CLKCNT          ;ADD ONE TO CLOCK COUNT OF WORD
023314  032767  000200  155644                  BIT     #BIT7,ONCEE     ;TRANSFERRING LAST WORD?
023322  001026                                  BNE     LSTWD           ;YES
023324  032767  000400  155634                  BIT     #BIT8,ONCEE     ;TRANSFERRING CRC WORD?
023332  001040                                  BNE     CRCWD           ;YES
023334  022767  000010  155644                  CMP     #8.,CLKCNT      ;CLOCK COUNT 8 OR GREATER?
023342  101401                                  BLOS    1$              ;YES
023344  000414                                  BR      2$              ;GET OUT
023346  022767  000021  155632          1$:     CMP     #17.,CLKCNT     ;CLOCK COUNT 17 OR GREATER?
023354  101410                                  BLOS    2$              ;YES GET OUT

023356  052701  004000                          BIS     #BIT11,GOOD     ;SET SB BIT
023362  022767  000017  155616                  CMP     #15.,CLKCNT     ;SHOULD LSR BE CLEARED
023370  001002                                  BNE     2$              ;NO
023372  042701  002000                          BIC     #BIT10,GOOD     ;CLEAR LSR
023376  000207                          2$:     RTS     PC              ;RETURN

                                        ;CALCULATE MR FOR LAST DATA WORD
023400  022767  000016  155600          LSTWD:  CMP     #14.,CLKCNT     ;IS THIS CLOCK 14 OR LESS?
023406  103011                                  BHIS    2$              ;YES GETOUT
023410  022767  000017  155570                  CMP     #15.,CLKCNT     ;IS THIS CLOCK 15?
023416  001003                                  BNE     1$              ;NO
023420  042701  002000                          BIC     #BIT10,GOOD     ;YES CLEAR LSR
023424  000402                                  BR      2$              ;GET OUT
023426  042701  020000                  1$:     BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
023432  000207                          2$:     RTS     PC

                                        ;CALCULATE MR FOR CRC WORD

023434  042701  020000                  CRCWD:  BIC     #BIT13,GOOD     ;CLEAR CRCW BIT
023440  022767  000017  155540                  CMP     #17,CLKCNT      ;IS THIS CLOCK 17"
023446  001002                                  BNE     1$              ;NO
023450  042701  002000                          BIC     #BIT10,GOOD     ;CLEAR LSR BIT
023454  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

023456  012767  000200  155516  GENCRC: MOV     #128.,REPT      ;128 WORDS PER SECTOR
023464  032767  000040  155452          BIT     #BITS,FLAG2     ;IN CRC TEST?
023472  001403                          BEQ     13$             ;NO
023474  012767  000220  155500          MOV     #144.,REPT      ;YES
023502  012705  026610          13$:    MOV     #INBUF,R5       ;GET STARTING ADDR OF OUTPUT BUFFER
023506  011504                          MOV     (R5),R4         ;GET DATA WD
023510  005067  155502                  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)

023514  012767  000022  155462  1$:     MOV     #18.,REPT1      ;GET 18 BITS PER WD
023522  032767  000040  155414          BIT     #BITS,FLAG2     ;IN CRC TEST?
023530  001403                          BEQ     2$              ;NO
023532  012767  000020  155444          MOV     #16.,REPT1      ;YES
023540  016767  155452  155432  2$:     MOV     WORK0,WCRC      ;SAVE CURRENT CRC WD
023546  005067  155440                  CLR     WK15            ;CLEAR BIT 15 FROM CRC AT T 1
023552  000241                          CLC                     ;CLEAR CARY
023554  006167  155436                  ROL     WORK0           ;SHIFT CRC WD LEFT
023560  006167  155426                  ROL     WK15            ;CONTAINS BIT 15 OF CRC
023564  032767  000040  155352          BIT     #BITS,FLAG2     ;IN CRC TEST?
023572  001004                          BNE     12$             ;YES
023574  022767  000021  155402          CMP     #17.,REPT1      ;DONE BITS 16 AND 17 YET?
023602  101406                          BLOS    3$              ;NO
023604  005067  155400          12$:    CLR     INBIT           ;CLEAR WORK LOC
023610  006104                          ROL     R4              ;PUT DATA BIT FROM BUFFER
023612  006167  155372                  ROL     INBIT           ;IN WORK1 LOC
023616  000402                          BR      4$
023620  005067  155364          3$:     CLR     INBIT           ;FOR BITS 16 AND 17
023624  016767  155362  155362  4$:     MOV     WK15,WORK       ;GET BIT 15 OF CRC
023632  004767  000220          5$:     JSR     PC,XXOR         ;XOR BIT15 WITH INPUT BIT
023636  042767  000001  155352          BIC     #BIT0,WORK0
023644  005767  155340                  TST     INBIT           ;TEST RESULT OF XOR
023650  001403                          BEQ     6$
023652  052767  000001  155336          BIS     #BIT0,WORK0
023660  016767  155324  155270  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

023666  005067  155322                  CLR     WORK
023672  032767  000002  155300          BIT     #BIT1,WCRC
023700  001403                          BEQ     7$
023702  052767  000001  155304          BIS     #BIT0,WORK
023710  016767  155242  155272  7$:     MOV     RS0,INBIT
023716  004767  000134                  JSR     PC,XXOR
023722  042767  000004  155266          BIC     #BIT2,WORK0
023730  005767  155254                  TST     INBIT             ;TEST RESULT OF XOR
023734  001403                          BEQ     8$
023736  052767  000004  155252          BIS     #BIT2,WORK0

                                ;FROM B0 IN WORK0 AND B14 IN WCRC CLACULATE BIT15 IN WORK0

023744  005067  155244          8$:     CLR     WORK
023750  032767  040000  155222          BIT     #BIT14,WCRC
023756  001403                          BEQ     9$
023760  052767  000001  155226          BIS     #BIT0,WORK
023766  016767  155164  155214  9$:     MOV     RS0,INBIT
023774  004767  000056                  JSR     PC,XXOR
024000  042767  100000  155210          BIC     #BIT15,WORK0
024006  005767  155176                  TST     INBIT             ;TEST RESULT OF XOR
024012  001403                          BEQ     10$
024014  052767  100000  155174          BIS     #BIT15,WORK0
024022  005367  155156          10$:    DEC     REPT1             ;DONE WITH WD
024026  001244                          BNE     2$                ;NO
024030  005367  155146                  DEC     REPT              ;DONE WITH SECTOR?
024034  001404                          BEQ     11$               ;YES
024036  062705  000002                  ADD     #2,R5             ;GET NEXT WD
024042  011504                          MOV     (R5),R4           ;GET DATA WD
024044  000623                          BR      1$
024046  016767  155144  155140  11$:    MOV     WORK0,WORK        ;SAVE CRC WORD IN WORK
024054  000207                          RTS     PC                ;EXIT

                                ;XOR SUBROUTINE

024056  016703  155132          XXOR:   MOV     WORK,R3
024062  046703  155122                  BIC     INBIT,R3
024066  046767  155122  155114          BIC     WORK,INBIT
024074  050367  155110                  BIS     R3,INBIT
024100  000207                          RTS     PC
```

```
                                        .SBTTL          $TYPE - TTY TYPEOUT ROUTINE

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

024102  010446                          $TYPE:  MOV     R4,-(6)                 ;SAVE R4
024104  010546                                  MOV     R5,-(6)                 ;SAVE R5
024106  017605  000004                          MOV     @4(6),R5                ;GET ADDRESS TO BE TYPED
024112  032705  177400                          BIT     #177400,R5              ;IS IT A TYPEM?
024116  001002                                  BNE     1$                      ;NO
024120  016605  000004                          MOV     4(6),R5                 ;GET ADDRESS OF CHARACTER
024124  105715                          1$:     TSTB    (R5)                    ;TERMINATOR?
024126  001423                                  BEQ     2$                      ;GET OUT IF SO
024130  122715  000012                          CMPB    #12,(R5)                ;IS THE CHAR A LINE FEED
024134  001012                                  BNE     4$                      ;NO - GET OUT
024136  116704  154653                          MOVB    FILCHR+1,R4             ;GET THE FILL COUNT
024142  116777  154646  154650          5$:     MOVB    FILCHR,@$TPB            ;TYPE A FILLER
024150  105777  154642                          TSTB    @TPS                    ;DONE YET?
024154  100375                                  BPL     .-4                     ;NO - WAIT
024156  005304                                  DEC     R4                      ;DEC COUNT
024160  001370                                  BNE     5$                      ;LOOP UNTIL 0
024162  112577  154632                  4$:     MOVB    (R5)+,@TPB              ;LOAD AND TYPE THE CHARACTER
024166  105777  154624                          TSTB    @TPS                    ;IS THE PRINTER READY
024172  100375                                  BPL     .-4                     ;WAIT UNTIL IT IS
024174  000753                                  BR      1$                      ;GET THE NEXT CHARACTER
024176  017646  000004                  2$:     MOV     @4(6),-(6)              ;GET ADDRESS TO BE TYPED
024202  062766  000002  000006                  ADD     #2,6(6)                 ;ADD 2 TO THE ADDRESS
024210  022666  000004                          CMP     (6)+,4(6)               ;IS IT .+2?
024214  001006                                  BNE     3$                      ;NO
024216  062705  000002                          ADD     #2,R5                   ;ADD 2 TO THE ADDRESS
024222  042705  000001                          BIC     #1,R5                   ;BACK UP TO AN EVEN BYTE
024226  010566  000004                          MOV     R5,4(6)                 ;RESTORE ADDRESS
024232  012605                          3$:     MOV     (6)+,R5                 ;RESTORE R5
024234  012604                                  MOV     (6)+,R4                 ;RESTORE R4
024236  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:"

024240  032737  000400  177570  .SCOPE: BIT     #SW8,@#SWR      ;LOOP ON SPEC. TEST?
024246  001404                          BEQ     1$              ;NO LOOP ON SPEC. TEST
024250  123767  177570  154522          CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   #SW7-0#
024256  001453                          BEQ     .OVER           ;NOT RIGHT TEST
024260  032737  040000  177570  1$:     BIT     #SW14,@#SWR     ;LOOP ON TEST?
024266  001045                          BNE     .KIT            ;LOOP ON TEST IS SET
024270  000416                          BR      3$              ;SKIP - NOP FOR XOR TESTER
024272  013746  000004                  MOV     @#4,-(6)        ;PUSH @#4 ON STACK
024276  012737  024316  000004          MOV     #4$,@#4         ;SET FOR TIMEOUT
024304  005737  177060                  TST     @#177060        ;ERROR ON XOR?
024310  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
024314  000422                          BR      .SVLAD          ;NO ERROR - GO TO NEXT TEST
024316  022626                  4$:     CMP     (6)+,(6)+       ;CLEAR STACK
024320  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
024324  000426                          BR      .KIT            ;ERROR - LOOP ON TEST
024326  032737  004000  177570  3$:     BIT     #SW11,@#SWR     ;KILL ITERATIONS
024334  001012                          BNE     .SVLAD          ;YES - KILL ITERATIONS
024336  105767  154437                  TSTB    ICNT+1          ;FIRST ONE?
024342  001404                          BEQ     2$              ;BRANCH IF FIRST
024344  126767  000060  154427          CMPB    TIMES,ICNT+1    ;DONE?
024352  003013                          BGT     .KIT            ;BRANCH IF NOT
024354  112767  000001  154417  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
024362  105267  154412          .SVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
024366  011667  154416                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
024372  016737  154402  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
024400  000002                          RTI                     ;RETURN

024402  105267  154373          .KIT:   INCB    ICNT+1          ;INC THE ITERATION COUNT
024406  016737  154366  177570  .OVER:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
024414  005767  154370                  TST     LAD             ;FIRST ONE?
024420  001760                          BEQ     .SVLAD          ;YES
024422  016716  154362                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
024426  000002                          RTI                     ;FIXES PS

024430  000001                  TIMES:  1                       ;RUN 1 TIMES
```

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

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

024432  032737  002000  177570  .HLT:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
024440  001402                          BEQ     1$              ;NO - SKIP
024442  104402  000007                  TYPE    ,BELL           ;RING BELL
024446  005267  154330          1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
024452  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
024460  001025                          BNE     2$              ;SKIP TYPEOUTS
024462  104402  024466                  TYPE    ,.+2            ;.ASCIZ <15><12>
024472  011667  154314                  MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
024476  162767  000002  154306          SUB     #2,HLTADR       ;FUDGE ADDRESS
024504  117767  154302  000066          MOVB    @HLTADR,.HLTCT  ;GET HLT ARGUEMENT
024512  016746  154274                  MOV     HLTADR,-(6)     ;PUT HLTADR ON STACK
024516  104404                          TYPEO                   ;TYPE STACK IN OCTAL
024520  104402  024524                  TYPE    ,.+2            ;.ASCIZ "  "
024530  004767  001152                  JSR     PC,RSREG        ;GO TO USER ERROR ROUTINE
024534  005737  177570          2$:     TST     @#SWR           ;HALT ON ERROR
024540  100001                          BPL     .+4             ;SKIP IF CONTINUE
024542  000000                          HALT                    ;HALT ON ERROR!
024544  032737  001000  177570          BIT     #SW9,@#SWR      ;CHECK FOR INHIBIT LOOP ON ERROR
024552  001010                          BNE     4$              ;SKIP IF LOOP ON ERROR
024554  105067  154221                  CLRB    ICNT+1          ;CLEAR ITERATION COUNT
024560  022737  021200  000042          CMP     #$ENDAD,@#42    ;ACT11 AUTO ACCEPT?
024566  001001                          BNE     3$              ;NO BR
024570  000000                          HALT
024572  000002                  3$:     RTI                     ;RETURN
024574  000167  177602          4$:     JMP     .KIT            ;LOOP ON TEST UNTIL NO ERRORS

024600  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.

024602  012767  170101  000160  .TYPEB: MOV     #170101,.PR             ;SET BIT FLAG AND 16. CHARACTER COUNT
024610  000411                          BR      .PTIT                   ;NOW TYPE IT IN BIT FORM
024612  112767  000001  000150  .TYPEO: MOVB    #1,.PR                  ;SET ZERO FILL SWITCH
024620  000402                          BR      .+6                     ;SKIP
024622  005067  000142          .TYPES: CLR     .PR                     ;SUPRESS LEADING ZERO'S
024626  112767  177772  000135          MOVB    #-6,.PR+1               ;SET COUNT
024634                          .PTIT:
024634  010446                          MOV     R4,-(6)                 ;PUSH R4 ON STACK
024636  010546                          MOV     R5,-(6)                 ;PUSH R5 ON STACK
024640  016605  000010                  MOV     10(6),R5                ;GET THE DATA
024644  012704  024772                  MOV     #.PR+2,R4               ;SET POINTER TO FIRST ASCII CHAR.
024650  105014                          CLRB    (4)                     ;CLEAR FIRST BYTE
024652  000411                          BR      .PRF                    ;ROTATE FIRST BIT
024654  105014                  .PRL:   CLRB    (4)                     ;CLEAR BYTE OF CHARACTER
024656  032767  000100  000104          BIT     #100,.PR                ;BIT TYPING MODE?
024664  001004                          BNE     .PRF                    ;YES - SKIP 2 ROTATES
024666  006105                          ROL     R5                      ;ROTATE BIT INTO C
024670  106114                          ROLB    (4)                     ;PACK IT
024672  006105                          ROL     R5                      ;ROTATE BIT INTO C
024674  106114                          ROLB    (4)                     ;PACK IT
024676  006105                  .PRF:   ROL     R5                      ;ROTATE BIT INTO C
024700  106114                          ROLB    (4)                     ;PACK IT
024702  105714                          TSTB    (4)                     ;IS IT ZERO?
024704  001402                          BEQ     .+6                     ;SKIP INC
024706  105267  000056                  INCB    .PR                     ;SET FILL SWITCH
024712  105767  000052                  TSTB    .PR                     ;CHECK FILL SWITCH
024716  001402                          BEQ     .+6                     ;SKIP BITSET
024720  152724  000060                  BISB    #'0,(4)+                ;MAKE INTO ASCII CHAR
024724  105267  000041                  INCB    .PR+1                   ;INC COUNT
024730  001351                          BNE     .PRL                    ;REPEAT
024732  022704  024772                  CMP     #.PR+2,R4               ;EMPTY BUFFER?
024736  001002                          BNE     .+6                     ;SKIP IF NOT
024740  112724  000060                  MOVB    #'0,(4)+                ;LOAD 1 ZERO
024744  105014                          CLRB    (4)                     ;NULL TERMINATOR
024746  104402  024772                  TYPE    ,.PR+2                  ;TYPE IT
024752  012605                          MOV     (6)+,R5                 ;POP STACK INTO R5
024754  012604                          MOV     (6)+,R4                 ;POP STACK INTO R4
024756  016666  000002  000004          MOV     2(6),4(6)               ;GET RID OF
024764  012616                          MOV     (6)+,(6)                ;DATA WORD
024766  000002                          RTI                             ;RETURN

024770  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'.

025014  012777  025142  000126  .POWER: MOV     #.ILLUP,@.PUVEC ;SET FOR FAST UP
025022  012777  000340  000122          MOV     #340,@.PUVEC$+2 ;PRIO:7
025030  010046                          MOV     R0,-(6)         ;PUSH R0 ON STACK
025032  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
025034  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
025036  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
025040  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
025042  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
025044  010667  000076                  MOV     SP,.SAVR6       ;SAVE SP
025050  012777  025060  000072          MOV     #.POWUP,@.PUVEC ;SET UP VECTOR
025056  000000                          HALT                    ;WAIT FOR PF

025060  016706  000062          .POWUP: MOV     .SAVR6,SP       ;GET SP
025064  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
025066  005201                  1$:     INC     R1              ;WAIT FOR THE INC
025070  001376                          BNE     1$              ;OF  WORD
025072  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
025074  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
025076  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
025100  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
025102  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
025104  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
025106  012737  025014  000024          MOV     #.POWER,@#24    ;SET UP THE POWER DOWN VECTOR
025114  012737  000340  000026          MOV     #340,@#26       ;PRIO:7
025122  104402  025126                  TYPE    ,.+2            ;.ASCIZ <15><12>"POWER"
025136  000167  174054                  JMP     MULSYS          ;JMP TO USER ADDRESS

025142  000000                  .ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
025144  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE

025146  000000                  .SAVR6: 0                       ;PUT THE SP HERE
025150  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.

025154  011646                          .RDOCT: MOV     (6),-(6)                ;MOVE THE PC
025156  016666  000004  000002                  MOV     4(6),2(6)               ;MOVE THE PS
025164  010146                                  MOV     R1,-(6)                 ;PUSH R1 ON STACK
025166  010246                                  MOV     R2,-(6)                 ;PUSH R2 ON STACK
025170  010346                                  MOV     R3,-(6)                 ;PUSH R3 ON STACK
025172  104412                          4$:     RDLIN                           ;READ A LINE INTO INPUT
025174  005001                                  CLR     R1                      ;INIT DATA WORD
025176  012703  025376                          MOV     #INPUT,R3               ;INIT POINTER
025202  112302                          1$:     MOVB    (3)+,R2                 ;GET A BYTE
025204  001417                                  BEQ     2$                      ;GET OUT IF ZERO
025206  122702  000060                          CMPB    #'0,R2                  ;CHECK FOR 0 OR GREATER
025212  003022                                  BGT     3$                      ;ERROR - LESS THAN 0
025214  122702  000067                          CMPB    #'7,R2                  ;CHECK FOR 7 OR LESS
025220  002417                                  BLT     3$                      ;ERROR - GREATER THAN 7
025222  006002                                  ROR     R2                      ;GET
025224  006002                                  ROR     R2                      ;INTO
025226  006002                                  ROR     R2                      ;POSITION
025230  006101                                  ROL     R1                      ;FIRST BIT
025232  006102                                  ROL     R2                      ;GET
025234  006101                                  ROL     R1                      ;SECOND BIT
025236  006102                                  ROL     R2                      ;GET
025240  006101                                  ROL     R1                      ;THIRD BIT
025242  000757                                  BR      1$                      ;LOOP
025244  010166  000012                  2$:     MOV     R1,12(6)                ;SAVE THE RESULT
025250  012603                                  MOV     (6)+,R3                 ;POP STACK INTO R3
025252  012602                                  MOV     (6)+,R2                 ;POP STACK INTO R2
025254  012601                                  MOV     (6)+,R1                 ;POP STACK INTO R1
025256  000002                                  RTI                             ;RETURN

025260                                  3$:
025260  104402  025264                          TYPE    ,.+2                    ;.ASCIZ "?"<15><12>
025270  000740                                  BR      4$                      ;TRY AGAIN
```

```
                                        .SBTTL          $RDLIN - TTY INPUT ROUTINE

                                        ;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
                                        ;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
                                        ;INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYPING
                                        ;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.

025272  010546                          .RDLIN: MOV     R5,-(6)                 ;SAVE R5
025274  012705  025376                  1$:     MOV     #INPUT,R5               ;GET ADDRESS
025300  022705  025416                  2$:     CMP     #INPUT+16.,R5           ;BUFFER FULL?
025304  001412                                  BEQ     4$                      ;YES - TYPE "?"
025306  105737  177560                          TSTB    @#177560                ;WAIT FOR
025312  100375                                  BPL     .-4                     ;A CHARACTER
025314  113715  177562                          MOVB    @#177562,(5)            ;GET CHARACTER
025320  142715  000200                          BICB    #200,(5)                ;GET RID OF JUNK
025324  122715  000177                          CMPB    #177,(5)                ;IS IT A RUBOUT
025330  001005                                  BNE     3$                      ;SKIP IF NOT
025332                                  4$:
025332  104402  025336                          TYPE    ,.+2                    ;.ASCIZ "?"<15><12>
025342  000754                                  BR      1$                      ;ZAP THE BUFFER AND LOOP
025344  111527  000000                  3$:     MOVB    (5),#0                  ;SET UP FOR TYPING
025350  104402  025346                          TYPE    ,3$+2                   ;ECHO IT
025354  122725  000015                          CMPB    #15,(5)+                ;CHECK FOR RETURN
025360  001347                                  BNE     2$                      ;LOOP IF NOT RETURN
025362  105065  177777                          CLRB    -1(5)                   ;ZAP RETURN (THE 15)
025366  104402  000012                          TYPE    ,12                     ;TYPE A LINE FEED
025372  012605                                  MOV     (6)+,R5                 ;RESTORE R5
025374  000002                                  RTI                             ;RETURN

025376  000020                          INPUT:  .BLKB   16.                     ;TTY INPUT AREA
```

```
4888                                    .SBTTL          $TRAP - TRAP HANDLER
4889
4890                                    ;THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
4891                                    ;SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
4892                                    ;THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
4893                                    ;FOLLOW THIS MACRO.
4894
4895 025416  011646                     .TRAP:  MOV     (6),-(6)           ;GET ADDRESS OF TRAP +2
4896 025420  162716  000002                     SUB     #2,(6)             ;MAKE IT ADDRESS OF TRAP
4897 025424  017616  000000                     MOV     @(6),(6)           ;GET TRAP INSTRUCTION
4898 025430  062716  121036                     ADD     #.TRP+2-TRAP,(6);GET DATA AND MAKE IT AN OFFSET
4899 025434  013607                     .TRP:   MOV     @(6)+,PC           ;GO TO PROPER SUBROUTINE
4900
4901 025436  024240                             .SCOPE                     ;SCOPE   =   TRAP+0      (104400)
4902 025440  024102                             .TYPE                      ;TYPE    =   TRAP+2      (104402)
4903 025442  024612                             .TYPEO                     ;TYPEO   =   TRAP+4      (104404)
4904 025444  024622                             .TYPES                     ;TYPES   =   TRAP+6      (104406)
4905 025446  025154                             .RDOCT                     ;RDOCT   =   TRAP+10     (104410)
4906 025450  025272                             .RDLIN                     ;RDLIN   =   TRAP+12     (104412)
4907 025452  025510                             .CLRDK                     ;CLRDK   =   TRAP+14     (104414)
4908 025454  025536                             .MRDMD                     ;MPDMD   =   TRAP+16     (104416)
4909 025456  021540                             .MRCK                      ;MRCK    =   TRAP+20     (104420)
4910 025460  021510                             .MRCLK                     ;MRCLK   =   TRAP+22     (104422)
4911 025462  021502                             .MRINT                     ;MRINT   =   TRAP+24     (104424)
4912 025464  021604                             .DSCK                      ;DSCK    =   TRAP+26     (104426)
4913 025466  021566                             .MRIND                     ;M IND   =   TRAP+30     (104430)
4914 025470  021632                             .XBIT                      ;XBIT    =   TRAP+32     (104432)
4915 025472  022032                             .CLKD1                     ;CLKD1   =   TRAP+34     (104434)
4916 025474  022126                             .CLKD2                     ;CLKD2   =   TRAP+36     (104436)
4917 025476  023010                             .CLKR1                     ;CLKR1   =   TRAP+40     (104440)
4918 025500  023240                             .CLKR2                     ;CLKR2   =   TRAP+42     (104442)
4919 025502  022640                             .RBIT                      ;RBIT    =   TRAP+44     (104444)
4920 025504  025630                             .GETSP                     ;GETSP   =   TRAP+46     (104446)
4921 025506  025670                             .SPASS                     ;SPASS   =   TRAP+50     (104450)
```

```
                                        ;CLEAR ALL DISK REGISTERS
025510  012777  000040  153364  .CLRDK: MOV     #40,@RSCS2              ;CLEAR ALL DSK REG
025516  016777  153436  153356          MOV     UNNUM,@RSCS2            ;GET UNIT NUMBER
025524  005067  153444                  CLR     MCCNT                   ;CLEAR MAINT CLOCK COUNT
025530  005067  153442                  CLR     MCCNT+2
025534  000002                          RTI

025536  012777  000001  153360  .MRDMD: MOV     #1,@RSMR                ;PUT DRIVE INTO MAINT MODE
025544  000002                          RTI

025546  005067  153442          WAITRY: CLR     WORK                    ;CLEAR COUNTER
025552  105777  153322          1$:     TSTB    @RSCS1                  ;TEST READY
025556  100406                          BMI     2$                      ;OK CONT
025560  005267  153430                  INC     WORK                    ;UPDATE COUNTER
025564  005767  153424                  TST     WORK                    ;DONE YET?
025570  001403                          BEQ     3$                      ;READY DID NOT COME UP
025572  000767                          BR      1$                      ;CONTINUE WAITING
025574  062716  000002          2$:     ADD     #2,(SP)                 ;UPDATE RETURN PC
025600  000207                  3$:     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

025602  012702  026610          MDATA:  MOV     #INBUF,R2               ;GET LEFT ADDRESS OF
025606  062702  000442                  ADD     #442,R2                 ;DATA TABLE
025612  012703  000220                  MOV     #220,R3                 ;SETUP COUNTER FOR 200 WORDS
025616  000241                          CLC                             ;CLEAR CARRY
025620  006142                  1$:     ROL     -(R2)                   ;SHIFT DATA PATTERN
025622  005303                          DEC     R3                      ;DO ALL
025624  001375                          BNE     1$                      ;WORDS
025626  000207                          RTS     PC

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

025630  012767  002001  153344  .GETSP: MOV     #1025.,REPT             ;SETUP COUNTER
025636  104430                          MRIND                           ;SEND INDEX PULSE TO MR REG
025640  104422                  1$:     MRCLK                           ;CLOCK MR
025642  005367  153334                  DEC     REPT                    ;TO REACH
025646  001374                          BNE     1$                      ;SECTOR PULSE
025650  032777  000400  153246          BIT     #400,@RSMR              ;DID SECTOR PULSE SET?????
025656  001401                          BEQ     2$                      ;YES
025660  000002                          RTI                             ;NO REPORT ERROR
025662  062716  000002          2$:     ADD     #2,(SP)                 ;UPDATE RETURN ADDR
025666  000002                          RTI

025670  104422                  .SPASS: MRCLK                           ;CLOCK PAST SECTOR PULSE
025672  104422                          MRCLK
025674  005067  153274                  CLR     MCCNT                   ;RESET MAINT CLOCK COUNTERS
025700  005067  153272                  CLR     MCCNT+2
025704  000002                          RTI
```

```
                                        ;ERROR TYPTXTOUT ROUTINE

025706  005767  176666          RSREG:  TST     .HLTCT                  ;SHOULD WE TYPTXT GOOD AND BAD
025712  001022                          BNE     8$                      ;NO
025714  104402  025720                  TYPE    ,.+2                    ;.ASCIZ " BAD="
025726  010046                          MOV     BAD,-(6)                ;PUT BAD ON STACK
025730  104404                          TYPEO                           ;TYPE STACK IN OCTAL
025732  104402  025736                  TYPE    ,.+2                    ;.ASCIZ " GOOD="
025746  010146                          MOV     GOOD,-(6)               ;PUT GOOD ON STACK
025750  104404                          TYPEO                           ;TYPE STACK IN OCTAL
025752  000402                          BR      8$                      ;TYPEOUT REGISTERS
025754  000167  000432                  JMP     PTDONE                  ;GET OUT
025760                          8$:
025760  104402  025764                  TYPE    ,.+2                    ;.ASCIZ " CS1="
025772  017746  153102                  MOV     @RSCS1,-(6)             ;PUT @RSCS1 ON STACK
025776  104404                          TYPEO                           ;TYPE STACK IN OCTAL
026000                          1$:
026000  104402  026004                  TYPE    ,.+2                    ;.ASCIZ " ER="
026012  017746  153076                  MOV     @RSER, (6)              ;PUT @RSER ON STACK
026016  104404                          TYPEO                           ;TYPE STACK IN OCTAL
026020                          2$:
026020  104402  026024                  TYPE    ,.+2                    ;.ASCIZ " CS2="
026032  017746  153044                  MOV     @RSCS2,-(6)             ;PUT @RSCS2 ON STACK
026036  104404                          TYPEO                           ;TYPE STACK IN OCTAL
026040  032767  000200  176532          BIT     #200,.HLTCT             ;TYPTXT SECOND SET ?
026046  001076                          BNE     SEEC                    ;YES
026050  032767  000100  176522          BIT     #AS,.HLTCT              ;TYPTXT ER ?
026056  001410                          BEQ     3$                      ;NO
026060  104402  026064                  TYPE    ,.+2                    ;.ASCIZ " AS="
026072  017746  153020                  MOV     @RSAS,-(6)              ;PUT @RSAS ON STACK
026076  104404                          TYPEO                           ;TYPE STACK IN OCTAL
026100  032767  000020  176472  3$:     BIT     #BA,.HLTCT              ;TYPTXT BUS ASSRESS
026106  001410                          BEQ     4$                      ;NO
026110  104402  026114                  TYPE    ,.+2                    ;.ASCIZ " BA="
026122  017746  152760                  MOV     @RSBA.-(6)              ;PUT @RSBA ON STACK
026126  104404                          TYPEO                           ;TYPE STACK IN OCTAL
026130  032767  000004  176442  4$:     BIT     #DA,.HLTCT              ;TYPTXT DA ?
026136  001410                          BEQ     5$                      ;NO
026140  104402  026144                  TYPE    ,.+2                    ;.ASCIZ " DA="
026152  017746  152732                  MOV     @RSDA,-(6)              ;PUT @RSDA ON STACK
026156  104404                          TYPEO                           ;TYPE STACK IN OCTAL
026160  032767  000010  176412  5$:     BIT     #WC,.HLTCT              ;TYPTXT WC?
026166  001410                          BEQ     6$                      ;NO
026170  104402  026174                  TYPE    ,.+2                    ;.ASCIZ " WC="
026202  017746  152676                  MOV     @RSWC,-(6)              ;PUT @RSWC ON STACK
026206  104404                          TYPEO                           ;TYPE STACK IN OCTAL
026210  032767  000040  176362  6$:     BIT     #DS,.HLTCT              ;DRIVE STATUS
026216  001475                          BEQ     PTDONE                  ;NO
026220  104402  026224                  TYPE    ,.+2                    ;.ASCIZ " DS="
```

```
026232  017746  152654                   MOV     @RSDS,-(6)        ;PUT @RSDS ON STACK
026236  104404                           TYPEO                     ;TYPE STACK IN OCTAL
026240  000167  000146                   JMP     PTDONE            ;GET OUT
026244  042767  000200  176326  SEEC:    BIC     #200,.HLTCT       ;CLEAR COMMON BIT
026252  032767  000240  176320           BIT     #DT,.HLTCT        ;TYPTXT DRIVE TYPE?
026260  001410                           BEQ     9$                ;NO
026262  104402  026266                   TYPE    ,.+2              ;.ASCIZ " DT="
026274  017746  152626                   MOV     @RSDT,-(6)        ;PUT @RSDT ON STACK
026300  104404                           TYPEO                     ;TYPE STACK IN OCTAL
026302  032767  000210  176270  9$:      BIT     #DB,.HLTCT        ;TYPTXT DATA BUFFER
026310  001410                           BEQ     10$               ;NO
026312  104402  026316                   TYPE    ,.+2              ;.ASCIZ " DB="
026324  017746  152572                   MOV     @RSDB,-(6)        ;PUT @RSDB ON STACK
026330  104404                           TYPEO                     ;TYPE STACK IN OCTAL
026332  032767  000220  176240  10$:     BIT     #MR,.HLTCT        ;TYPTXT MN?
026340  001410                           BEQ     11$               ;NO
026342  104402  026346                   TYPE    ,.+2              ;.ASCIZ " MR="
026354  017746  152544                   MOV     @RSMR,-(6)        ;PUT @RSMR ON STACK
026360  104404                           TYPEO                     ;TYPE STACK IN OCTAL
026362  032767  000204  176210  11$:     BIT     #LA,.HLTCT        ;TYPTXT LA?
026370  001410                           BEQ     PTDONE            ;NO
026372  104402  026376                   TYPE    ,.+2              ;.ASCIZ " LA="
026404  017746  152510                   MOV     @RSLA,-(6)        ;PUT @RSLA ON STACK
026410  104404                           TYPEO                     ;TYPE STACK IN OCTAL
026412  052767  100000  152546  PTDONE:  BIS     #BIT15,ONCEE      ;SET FORND ERROR FLAG
026420  032767  000040  152540           BIT     #BIT5,ONCEE
026426  001466                           BEQ     1$
026430  104402  026434                   TYPE    ,.+2              ;.ASCIZ <15><12>"MAINT CLOCK COUNT "
026462  016767  152506  152536           MOV     MCCNT,WORK4       ;GET MAINT CLOCK COUNT
026470  016767  152502  152524           MOV     MCCNT+2,WORK2     ;CAL NUMBERS FOR DOUBLE PRECISION
026476  006167  152520                   ROL     WORK2
026502  006167  152520                   ROL     WORK4
026506  000241                           CLC
026510  016746  152512                   MOV     WORK4,-(6)        ;PUT WORK4 ON STACK
026514  104406                           TYPES                     ;TYPE STACK IN OCTAL - SUPRESS
026516  012767  000005  152504           MOV     #5,WORK5
026524  005067  152502          2$:      CLR     WORK6
026530  006167  152466                   ROL     WORK2
026534  006167  152472                   ROL     WORK6
026540  006167  152456                   ROL     WORK2
026544  006167  152462                   ROL     WORK6
026550  006167  152446                   ROL     WORK2
026554  006167  152452                   ROL     WORK6
026560  016746  152446                   MOV     WORK6,-(6)        ;PUT WORK6 ON STACK
026564  104406                           TYPES                     ;TYPE STACK IN OCTAL - SUPRESS
026566  005367  152436                   DEC     WORK5
026572  001354                           BNE     2$
026574  104402  026600                   TYPE    ,.+2              ;.ASCIZ <15><12>
026604  000207                  1$:      RTS     PC
026606  000000                  FUDGE:   .WORD   0                 ;THIS WORD IS INSERTED TO INSURE INBUF
                                                                   ;STARTS ON BOUNDARY OF 4,10 SEE AIDS PR#2507.

026610  000300                  INBUF:   .BLKW   300
027410  000300                  OUTBUF:  .BLKW   300
        000001                           .END
```

```
AOECK     020320           4035#
AS      = 000100            809#    1065    2026    2030    2033    2133    2137    2140    2394    3525    4999
ATA     = 100000            830#
BA      = 000020            807#    1035    1093    1099    1181    1185    1189    1211    1397    1401    5004
BAI     = 000010            833#    4123
BEGIN     001234            718      893#
BEGIN1    000232            714      720#
BEGIN2    000226            718#     721
BELL    = 000007            731#    4149    4708
BITBA     004424           1393#
BITCS2    004310           1369#
BITST     004236           1356#
BITWC     004354           1381#
BIT0    = 000001            740#     981     999    3523    4458    4460    4463    4568    4571    4579    4592
BIT1    = 000002            741#    1819    1841    1845    1850    1852    1860    1862    1867    1870    2653    4577
BIT10   = 002000            750#    1740    1754    1771    4510    4518    4528
BIT11   = 004000            751#    4073    4467    4507
BIT12   = 010000            752#    2368    2411    2414    2876    4288    4313    4315
BIT13   = 020000            753#     953     964     985    4520    4525
BIT14   = 040000            712      754#   4297    4318    4320    4590
BIT15   = 100000            755#     956     966     973    3607    4462    4595    4598    5046
BIT2    = 000004            742#    1897    1915    3588    3596    3600    3603    3606    3609    3612    4171    4173    4448
                           4487     4582    4585
BIT3    = 000010            743#    1903    1919    1952    1979    1981    1983    4084    4099    4443    4452
BIT4    = 000020            744#     912     975    3109    3144    3316    3342    3687    4164
BIT5    = 000040            745#    1818    1997    2046    2058    2103    2151    2256    2431    2479    2668    2728    2900
                           2950     3102    3163    3309    3360    3581    3682    3729    3776    4081    4415    4490    4536
                           4549     4557    5047
BIT6    = 000100            717      720     746#    904    2567    3864    4234    4241    4450
BIT7    = 000200            747#    2566    2578    2585    2591    3863    3875    3882    3888    4497
BIT8    = 000400            748#    2580    2590    2796    2798    3018    3020    3231    3233    3428    3430    3877    3887
                           4243     4413    4453    4468    4499
BIT9    = 001000            749#    1602    1604    1607    1745    1747    1749
CALRTB    023252           4439     4479    4484#
CHCKDV    002142            978      981#
CHG       021364           1963     1970    1977    4178#
CLKCNT    001206            854#    4242*   4412*   4496*   4501    4504    4508    4514    4516    4526
CLKD1   = 104434           2569     3866    4915#
CLKD2   = 104436           2573     3870    4916#
CLKR1   = 104440           2790     3012    3225    3422    4917#
CLKR2   = 104442           2792     3014    3227    3424    4918#
CLOOK     023020           4441#    4481
CLROK   = 104414           1123     1356    1402    1407    1435    1464    1641    1817    1854    1896    1937    1996    2045
                           2150     2186    2202    2237    2255    2288    2305    2335    2356    2430    2667    2899    3101
                           3308     3512    3560    3613    3631    3662    3681    3704    3726    3968    4047    4077    4097
                           4118     4907#
CRCTAB    022330           4333     4345#
CRCTYP    022220           3493     4333#
CRCWD     023434           4500     4525#
CS1     = 000001            803#    1030    1088    1128    1132    1136    1144    1361    1365    1443    1448
CS2     = 000200            810#    1021    1084    1154    1164    1168    1173    1418
DA      = 000004            805#    1041    1256    1260    1264    1285    2185    2236    2389    2405    2646    2652    2856
                           2862     3079    3085    3288    3485    3597    3604    3610    3697    3701    3707    3711    3947
                           3956     5009
DAO     = 001000            831#
DB      = 000210            812#    5031
```

```
DCK   = 100000    832#
DISPLA= 177570    730#   4687*   4691*
DLT   = 100000    825#
DONE    021134    994    4143#
DONEE   002316    976     982     994#
DRY   = 000200    826#
DS    = 000040    808#   1025    1843    1910    1914    1925    1929    2026    2030    2033    2133    2137    2140
                 2180    2189    2193    2231    2240    2244    2285    2291    2295    2332    2338    2342    2396
                 2856    3079    3288    3485    3491    3525    3543    3597    3604    3610    3616    3656    3660
                 3666    3670    3692    3697    3707    3711    3939    3953    4042    4045    4050    4053    4095
                 4110    4132    4136    5019
DSCK  = 104426   2008    2014    2021    2083    2089    2128    2158    2175    2209    2226    2263    2280    2312
                 2327    4005    4912#
DT    = 000240    814#   5026
DVNUM   001722    947     950#
ER    = 000002    804#   1047    1105    1293    1297    1304    1310    1317    3543
ERR   = 040000    829#
ERRORS  001002    762#   4709*   4732
E1910   022622   4373    4400#
E1915   022631   4375    4402#
E192    022606   4369    4398#
E197    022614   4371    4399#
E2410   022561   4363    4392#
E2412   022570   4365    4394#
E2415   022577   4367    4396#
E242    022537   4357    4386#
E245    022545   4359    4388#
E247    022553   4361    4390#
E3010   022512   4351    4380#
E3012   022521   4353    4382#
E3015   022530   4355    4384#
E302    022470   4345    4377#
E305    022476   4347    4378#
E307    022504   4349    4379#
FILCHR  001014    766#   4636    4637
FLAG2   001144    717*    720*    837#    902*    904    2429*   2653    2676*   2908*   3100*   3109    3144*   3307*
                 3316    3342*   3725*   4076*   4415    4448    4458    4460*   4463*   4536    4549    4557

FLOTBA  003346   1194#
FLOTDA  003546   1269#
FLOTWC  003506   1232#
FUDGE   026606   5071#
GENCRC  023456   2592    2800    3022    3894    4535#
GETSP = 104446   2465    2713    2936    3148    3346    3762    4920#
HLT   = 104000    726#    937    1021    1025    1030    1035    1041    1047    1053    1059    1065    1084    1088
                 1093    1099    1105    1110    1115    1128    1132    1136    1144    1154    1160    1164    1168
                 1173    1181    1185    1189    1200    1211    1219    1223    1227    1238    1248    1256    1260
                 1264    1275    1285    1293    1297    1304    1310    1317    1327    1334    1342    1350    1361
                 1365    1377    1385    1389    1397    1401    1418    1423    1428    1443    1448    1475    1481
                 1493    1497    1506    1510    1513    1522    1526    1535    1539    1549    1553    1559    1566
                 1570    1581    1595    1599    1615    1627    1654    1662    1665    1669    1672    1681    1685
                 1693    1704    1716    1728    1758    1780    1791    1798    1801    1842    1843    1851    1861
                 1869    1910    1914    1925    1929    1950    1964    1971    1978    2010    2016    2023    2026
                 2030    2033    2055    2062    2066    2073    2077    2085    2091    2098    2107    2111    2120
                 2124    2130    2133    2137    2140    2160    2174    2177    2180    2185    2189    2193    2211
                 2225    2228    2231    2236    2240    2244    2265    2279    2282    2285    2291    2295    2314
                 2326    2329    2332    2338    2342    2385    2389    2394    2398    2405    2408    2468    2474
```

```
                       2483    2487    2496    2500    2508    2512    2520    2524    2528    2534    2538    2547    2551
                       2558    2572    2577    2608    2616    2620    2629    2633    2640    2646    2649    2652    2716
                       2723    2732    2736    2745    2749    2756    2760    2766    2774    2779    2791    2793    2813
                       2817    2821    2825    2835    2839    2849    2856    2859    2862    2881    2939    2945    2954
                       2958    2967    2971    2979    2983    2989    2996    3001    3013    3015    3035    3039    3043
                       3047    3057    3061    3071    3079    3082    3085    3151    3158    3167    3171    3180    3184
                       3191    3195    3201    3209    3214    3226    3228    3244    3248    3252    3256    3266    3270
                       3280    3288    3291    3349    3355    3364    3368    3377    3381    3388    3392    3398    3406
                       3411    3423    3425    3441    3445    3449    3453    3463    3467    3477    3485    3488    3491
                       3525    3537    3543    3547    3554    3566    3597    3604    3610    3616    3652    3656    3660
                       3666    3670    3692    3697    3701    3707    3711    3765    3771    3780    3784    3793    3797
                       3805    3809    3817    3821    3825    3831    3835    3844    3848    3855    3869    3874    3908
                       3915    3919    3927    3931    3937    3939    3947    3950    3953    3956    4007    4016    4023
                       4027    4039    4042    4045    4050    4053    4092    4095    4107    4110    4132    4136
HLTADR  001012          765#   4713*   4714*   4715    4716    4733
IADONE  016446         3591    3618#
ICNT    001000          761#    900*   4163*   4665    4680    4682    4684*   4685*   4687    4690*   4691    4697    4725*
IE    = 000100          834#
ILFDON  006734         1887    1889#
INBIT   001210          855#   4561*   4563*   4565*   4569    4572    4580*   4583    4593*   4596    4612    4613*   4614*
INBUF   026610         2445    2462    2559    2594    2684    2783    2802    2868    2916    2933    3005    3024    3126
                       3218    3234    3325    3415    3431    3741    3759    3856    3896    4083*   4085    4100    4124
                       4125*   4539    4945    5074#
INFTST  021130         4075    4140#
INPUT   025376          910    4833    4866    4867    4887#
INT     004604         1435#
INTDON  004674         1444    1449#
INTMR   020024         3757    3945#
INTMR1  020074         3940    3957#
IR    = 000100          820#
LA    = 000204          811#   1481    1513    1581    1665    1672    5041
LAD     001010          764#    901*   3583*   4162*   4686*   4692    4694    4697
LBT   = 002000          828#
LSTEV   001146          838#   4236*   4250*   4295
LSTOD   001150          839#   4237*   4249*   4284
LSTWD   023400         4498    4514#
MCCNT   001174          850#   1644*   4013*   4204*   4205*   4445*   4446*   4925*   4926*   4970*   4971*   5050    5051
MDATA   025602         3294    3497    4945#
MPRO  = 172100          848#
MR    = 000220          813#   1053    1110    1327    1334    1342    1350    1481    1513    1581    2468    2716    2939
                       3151    3349    3765    5036
MRADE   020100         3968#
MRCAL   023310         4300    4323    4496#
MRCILF  006432         1854#
MRCK  = 104420         1467    1473    1491    1495    1504    1508    1520    1524    1533    1537    1547    1551    1557
                       1564    1568    1593    1597    1647    1652    1660    1667    1679    1683    1799    1825    1899
                       1939    1999    2048    2053    2060    2064    2071    2075    2096    2105    2109    2118    2122
                       2153    2204    2258    2307    2358    2434    2472    2481    2485    2494    2498    2506    2510
                       2518    2522    2526    2532    2536    2545    2549    2556    2606    2614    2618    2627    2631
                       2638    2671    2721    2730    2734    2743    2747    2754    2758    2764    2772    2777    2811
                       2815    2819    2823    2833    2837    2847    2903    2943    2952    2956    2965    2969    2977
                       2981    2987    2994    2999    3033    3037    3041    3045    3055    3059    3069    3105    3156
                       3165    3169    3178    3182    3189    3193    3199    3207    3212    3242    3246    3250    3254
                       3264    3268    3278    3312    3353    3362    3366    3375    3379    3386    3390    3396    3404
                       3409    3439    3443    3447    3451    3461    3465    3475    3514    3585    3636    3684    3732
                       3769    3778    3782    3791    3795    3803    3807    3815    3819    3823    3829    3833    3842
```

```
                          3846     3853     3906     3913     3917     3925     3929     3935     3971     4014     4021     4025     4037
                          4909#
MRCLK = 104422            1490     1494     1503     1507     1519     1523     1532     1536     1546     1550     1556     1563     1567
                          1589     1592     1596     1620     1659     1666     1678     1682     1699     1710     1723     1733     1775
                          1786     2013     2020     2059     2063     2070     2074     2079     2088     2104     2108     2117     2121
                          2127     2480     2484     2493     2497     2505     2509     2517     2521     2525     2531     2535     2544
                          2548     2555     2604     2613     2617     2626     2630     2637     2729     2733     2742     2746     2753
                          2757     2763     2810     2814     2818     2822     2832     2836     2846     2853     2951     2955     2964
                          2968     2976     2980     2986     3032     3036     3040     3044     3054     3058     3068     3076     3164
                          3168     3177     3181     3188     3192     3198     3241     3245     3249     3253     3263     3267     3277
                          3285     3361     3365     3374     3378     3385     3389     3395     3438     3442     3446     3450     3460
                          3464     3474     3482     3777     3781     3790     3794     3302     3806     3814     3818     3822     3828
                          3832     3841     3845     3852     3905     3912     3916     3424     3928     3934     4004     4020     4024
                          4035     4036     4198     4199     4279     4310     4910#    4959     4968     4969
MRCRC     014230          3101#    3295
MRCRC1    014434          3164#    3173
MRCRC2    014510          3188#    3197
MRCRC3    014612          3217#
MRCRC4    014772          3261#
MRCRCS    015036          3285#
MROCK     015110          3308#    3498
MROCK1    015314          3361#    3370
MROCK2    015370          3385#    3394
MROCK3    015472          3414#
MROCK4    015652          3458#
MROCKS    015716          3482#
MROMD = 104416            1824     1898     1938     1998     2047     2152     2203     2257     2306     2357     3584     4908#
MROSEL    011136          2356#
MRO3      012744          2782#
MRO4      013144          2831#
MRO5      013210          2853#
MRO6      013246          2868#
MREX      017112          3726#
MREX1     017306          3777#    3786
MREX2     017362          3802#    3811
MREX3     017446          3828#    3837
MRIFT     016032          3512#
MRILF     006214          1817#
MRIND = 104430            1466     1471     1646     1650     1792     1831     1942     2002     2052     2095     2433     2471     2670
                          2720     2902     2942     3104     3155     3311     3352     3513     3635     3642     3643     3644     3683
                          3731     3768     3970     3987     4012     4913#    4958
MRINT = 104424            1469     1649     1827     1901     1941     2001     2050     2155     2206     2260     2309     2360     2436
                          2673     2905     3107     3314     3516     3587     3638     3686     3734     3973     4911#
MRLF1     006240          1820     1824#    1888
MRLF2     006252          1832#
MRLF3     006652          1873#    1875     1877     1879     1881     1883     1885
MROP      006736          1896#
MROPER    007126          1937#    1982
MROPI     016450          3631#
MROPIA    016642          3659     3662#
MRPAR     016764          3681#
MRRD      012376          2667#
MRRD1     012566          2729#    2738
MRRD2     012642          2753#    2762
MRRT1     007716          2059#    2068
MRSRC     007652          2045#
```

```
MRSRCH  007470           1996#
MRTIME  004676           1464#
MRTIM1  004744           1490#   1499
MRT2    005022           1512    1517#
MRT2A   005032           1519#   1528
MRT2B   005066           1532#   1541
MRT2C   005122           1546#   1555
MRT3    005240           1588#   1618
MRT4    005432           1641#
MRT4A   005504           1659#   1674
MRT4B   005562           1682#   1768
MRT4C   005614           1699#   1706
MRT4D   005666           1722#   1760
MRT4E   005674           1723#   1730    1772
MRT4F   006114           1735    1774#
MRT4G   006146           1786#
MRVR    020404           4073#
MRVRR   020714           4109    4115#
MRVR1   020544           4094    4097#
MRVR2   020720           2655    4118#
MRWCK   013364           2899#
MRWCK1  013540           2951#   2960
MRWCK2  013614           2976#   2985
MRWCK3  013716           3004#
MRWCK4  014116           3053#
MRWCK5  014162           3076#
MRWRT   011456           2430#   4115
MRWRT1  011624           2480#   2489
MRWRT2  011700           2505#   2514
MRWRT3  011764           2531#   2540
MRWR1   010066           2104#   2113
MULSYS  021216           4154    4160#   4814
MULTII  001672            906     911     943#   4166
N     = 000065            725#   1002    1006#   1066    1070#   1116    1120#   1137    1138#   1146    1150#   1175    1176#
                         1191    1192#   1203    1204#   1212    1213#   1228    1229#   1242    1243#   1249    1250#   1265
                         1266#   1279    1280#   1286    1287#   1299    1300#   1305    1306#   1311    1312#   1318    1319#
                         1328    1329#   1335    1336#   1343    1344#   1351    1355#   1367    1368#   1379    1380#   1391
                         1392#   1403    1407#   1431    1435#   1450    1454#   1628    1632#   1802    1806#   1890    1894#
                         1931    1935#   1984    1988#   2034    2038#   2141    2145#   2194    2198#   2246    2250#   2296
                         2300#   2343    2347#   2416    2420#   2656    2660#   2889    2893#   3086    3090#   3297    3301#
                         3500    3504#   3569    3573#   3619    3623#   3671    3675#   3713    3717#   3958    3962#   4066
                         4070#
NED   = 010000            823#   2396
NEDOON  011356           2372    2411#
NNOO    011446           2404    2409    2412    2415#
NOPERR  021270           1909    1913    1924    1928    4171#
NOWEV   001152            840#   4246*   4250    4259*   4261*   4269*   4271*   4292    4316    4420*   4426*   4428*   4435*
                         4437*   4488
NOWGO   002322            940     974     993     999#   4167
NOWOO   001154            841#   4245*   4249    4256*   4258*   4266*   4268*   4281    4311    4419*   4423*   4425*   4432*
                         4434*   4485
ONCEE   001166            846#    903*    912*    956*    966     973*    975     981     999*   1465*   1602    1604*   1607*
                         1642*   1643*   1740    1745    1747*   1749*   1754*   1771*   1818*   1819    1841*   1845*   1850*
                         1852*   1860*   1862*   1867*   1870*   1897*   1903    1915*   1919*   1930*   1952    1979    1981*
                         1983*   1997*   2046*   2058*   2103*   2151*   2256*   2411    2414*   2431*   2432*   2479*   2566*
                         2567*   2578    2580    2585*   2590*   2591*   2668*   2669*   2728*   2796    2798*   2900*   2901*
```

```
                          2950*   3018    3020*   3102*   3103*   3163*   3231    3233*   3309*   3310*   3360*   3428    3430*
                          3581*   3588    3596*   3600*   3603*   3606*   3609*   3612*   3682*   3729*   3730*   3776*   3863*
                          3864*   3875    3877    3882*   3887*   3888*   4081*   4164    4171    4173*   4234    4241*   4243
                          4413    4453    4468    4497    4499    5046*   5047
OR      = 000200           821#
OUTBUF    027410          2705    2709    2869    3139    3143    3337    3341    3520    3550    3553    3632    3977    3980
                          5075#
PCNT      001004           763#   4144*   4145*   4156
PGE     = 002000           822#
PGTRAP    004660          1436    1445#
PIP     = 020000           827#
PS      = 177776           727#    728    1006*   1071*   1438*   3727*
PSW     = 177776           728#
PTDONE    026412          4984    5020    5024    5042    5046#
QQ      = 000001          1002#   1066#   1116#   1137#   1146#   1175#   1191#   1203#   1212#   1228#   1242#   1249#   1265#
                          1279#   1286#   1299#   1305#   1311#   1318#   1328#   1335#   1343#   1351#   1367#   1379#   1391#
                          1403#   1431#   1450#   1628#   1802#   1890#   1931#   1984#   2034#   2141#   2194#   2246#   2296#
                          2343#   2416#   2656#   2889#   3086#   3297#   3500#   3569#   3619#   3671#   3713#   3958#   4066#
RBIT    = 104444          2789    3011    3224    3421    4919#
RDLIN   = 104412           909    4831    4906#
RDOCT   = 104410           915    4905#
REGCHG    021400          4178    4183#
REPT      001202           852#   1489*   1498*   1518*   1527*   1531*   1540*   1545*   1554*   1588*   1590*   1619*   1621*
                          1658*   1673*   1698*   1705*   1721*   1759*   1770*   1774*   1781*   2012*   2017*   2057*   2067*
                          2078*   2080*   2087*   2092*   2102*   2112*   2478*   2488*   2504*   2513*   2530*   2539*   2612*
                          2621*   2727*   2737*   2752*   2761*   2786*   2794*   2808*   2831*   2840*   2870*   2878*   2949*
                          2959*   2975*   2984*   3008*   3016*   3030*   3053*   3062*   3162*   3172*   3187*   3196*   3221*
                          3229*   3239*   3261*   3271*   3359*   3369*   3384*   3393*   3418*   3426*   3436*   3458*   3468*
                          3775*   3785*   3801*   3810*   3827*   3836*   3911*   3920*   3981*   3983*   3991*   3999*   4019*
                          4028*   4086*   4090*   4101*   4105*   4455    4535*   4538*   4601*   4957*   4960*
REPT1     001204           853#   1587*   1617*   1722*   1729*   1769*   2785*   2799    2803*   3007*   3021    3025*   3220*
                          3417*   4464*   4466*   4470    4548*   4551*   4559    4599*
RMRC1     010210          2150#
RMRC2     010374          2202#
RMRC3     010550          2255#
RMRC4     010722          2305#
RSAS      001116           781#    933     935     950    1063    1946    1965    2027*   2028    2134*   2135    2318*   2392
                          3518*   500[illegible]
RSBA      001106           777#   100[illegible]*   1033    1074*   1091    1178*   1179    1182*   1183    1186*   1187    1196*   1197
                          1207*   120[illegible]*   1209    1393*   1395    1398*   1399    2462*   2705*   2933*   3139*   3337*   3520*
                          3550    3552    3632*   3759*   3977*   4085*   4100*   4124*   5007
RSBAB     001142           791#   1394*
RSCS1     001100           774#   1008*   1028    1073*   1086    1125*   1126    1129*   1130    1133*   1134    1140*   1141*
                          1142    1357*   1359    1362*   1363    1439*   1446    1832*   1905*   1921*   1954*   1957*   2007*
                          2024    2031    2082*   2131    2138    2157*   2164*   2178    2191    2208*   2229    2242    2262*
                          2283    2293    2311*   2330    2340    2374*   2387    2464*   2644    2707*   2854    2935*   3077
                          3141*   3286    3339*   3483    3522*   3523    3541    3593*   3607    3640*   3658    3695    3705
                          3761*   3945    3986*   4087*   4088    4093    4102*   4103    4108    4126*   4127    4134    4933
                          4987
RSCS1B    001134           788#   1358*   23[illegible]5*
RSCS2     001102           775#    929*    944*    948*    959*   1007*   1014*   1019    1022*   1072*   1079*   1081*   1082
                          1085*   1152    1155*   1156    1158    1161*   1162    1165*   1166    1169*   1170*   1171    1174*
                          1369*   1370*   1374    1413    2366*   2368    2376*   2380    2396    2400*   3532    3561    3687*
                          4084*   4099*   4123*   4129    4923*   4924*   4995
RSCS2B    001136           789#   1371*
RSDA      001110           778#   1010*   1039    1075*   1097    1253*   1254    1257*   1258    1261*   1262    1271*   1272
```

```
                         1281*   1282*   1283    2003*   2006*   2156*   2183    2207*   2215*   2234    2261*   2310*   2362*
                         2377*   2401    2650    2860    3083    3592*   3688*   3699    3954    3975*   5012
RSDB      001122          783#   1077*   1408*   1409*   1420    1425    5034
RSDS      001112          779#   1023    1833    1855    1911    1926    1972    3601    3654    3667    4043    4051    4220
                         5022
RSDT      001126          785#    960     962    1574    1600    1738    2004    5029
RSER      001114          780#    930*    945*   1011*   1045    1076*   1103    1290*   1291    1294*   1295    1301*   1302
                         1307*   1308    1313*   1314*   1315    1846    1863    1907    1922    1944*   1958    2168    2187
                         2219    2238    2269*   2273    2289    2321    2336    2367    3489    3517*   3594    3614    3647
                         3663    3690    3709    3893*   3951    4040    4048    4991
RSLA      001120          782#   1479    1511    1579    1611    1623    1663    1670    1689    1700    1712    1724    1737
                         1776    1787    1794    5044
RSMR      001124          784#   1012*   1051    1080*   1108    1322*   1323    1330*   1331*   1332    1337*   1338    1345*
                         1346    2770*   2771*   2775*   2776*   2992*   2993*   2997*   2998*   3205*   3206*   3210*   3211*
                         3402*   3403*   3407*   3408*   3995*   3996*   4202*   4203*   4208    4217*   4218*   4301    4324
                         4442*   4444*   4473    4929*   4962    5039
RSREG     025706         4719    4975#
RSVCPS    001132          787#   1437*   3758*
RSVEC     001130          786#   1436*   3757*
RSWC      001104          776#   1013*   1057    1078*   1113    1216*   1217    1220*   1221    1224*   1225    1234*   1235
                         1244*   1245*   1246    1381*   1383    1386*   1387    2463*   2647    2706*   2857    2934*   3080
                         3140*   3289    3338*   3486    3521*   3545    3633*   3760*   3948    3976*   4082*   4098*   4122*
                         5017
RSWCB     001140          790#   1382*
RSO       001156          842#   4572*   4580    4593
SAVEE     001172          849#   2799*   2803    3021*   3025    3111*   3236    3292*   3318*   3495*   4335
SC      = 100000          819#
SCOPE   = 104400         1005    1063    1119    1137    1149    1175    1191    1203    1212    1228    1242    1249    1265
                         1279    1286    1299    1305    1311    1318    1328    1335    1343    1354    1367    1379    1391
                         1406    1434    1453    1631    1805    1893    1934    1987    2037    2144    2197    2249    2299
                         2346    2419    2659    2892    3089    3300    3503    3572    3622    3674    3716    3961    4069
                         4143    4901#
SEEC      026244         4998    5025#
SPASS   = 104450         2469    2717    2940    3152    3350    3766    4921#
STTEST    002006          954     957#    980
SWR     = 177570          729#    730     953     964     985    2876    4073    4146    4663    4665    4667    4678    4706
                         4710    4720    4723
SW10    = 002000          704#   4146    4726
SW11    = 004000          703#   4678
SW12    = 010000          702#   4147    4149
SW13    = 020000          701#   4710
SW14    = 040000          700#   4667
SW15    = 100000          699#
SW8     = 000400          707#   4663
SW9     = 001000          706#   4723
TBOIA     020632         4096    4113#
TIMES     024430         1000    1001*   1124*   4682    4697#
TIMSV     001170          847#   1000*   1124
TPB       001020          768#   4637*   4642*
TPS       001016          767#   4638    4643
TRE     = 040000          818#
TRMR      021435         2172    2223    2277    4189#
TRY       001702          945#    949
TRYNX     002116          958     963     975#   4140
TSTEVB    022056         4282    4285    4292#
TST1      002344         1005#
```

```
TST10   003376               1203#
TST11   003422               1212#
TST12   003504               1228#
TST13   003536               1242#
TST14   003562               1249#
TST15   003644               1265#
TST16   003676               1279#
TST17   003722               1286#
TST2    002566               1069#
TST20   003764               1299#
TST21   004006               1305#
TST22   004030               1311#
TST23   004054               1318#
TST24   004112               1328#
TST25   004140               1335#
TST26   004176               1343#
TST27   004234               1354#
TST3    003002               1119#
TST30   004306               1367#
TST31   004352               1379#
TST32   004422               1391#
TST33   004474               1406#
TST34   004602               1434#
TST35   004674               1453#
TST36   005430               1631#
TST37   006212               1805#
TST4    003074               1137#
TST40   006734               1893#
TST41   007124               1934#
TST42   007466               1987#
TST43   007650               2037#
TST44   010206               2144#
TST45   010372               2197#
TST46   010546               2249#
TST47   010720               2299#
TST5    003122               1149#
TST50   011134               2346#
TST51   011446               2419#
TST52   012374               2659#
TST53   013362               2892#
TST54   014220               3089#
TST55   015100               3300#
TST56   016030               3503#
TST57   016250               3572#
TST6    003262               1175#
TST60   016446               3622#
TST61   016762               3674#
TST62   017102               3716#
TST63   020076               3961#
TST64   020402               4069#
TST7    003344               1191#
TTAGG   002570               1071#
TYPE  = 104402                908     914     931     955     968     972     989     990    1837    1859    1868    1949    1962
                             1969    1976    2172    2173    2223    2224    2277    2278    2325    2413    2883    3492    3651
                             3665    4114    4131    4148    4161    4174    4178    4341    4345    4347    4349    4351    4353
                             4355    4357    4359    4361    4363    4365    4367    4369    4371    4373    4375    4708    4712
```

```
                       4718    4773    4813    4856    4876    4879    4883    4902*   4977    4980    4986    4990    4994
                       5001    5006    5011    5016    5021    5028    5033    5038    5043    5049    5069
TYPEO  = 104404        4717    4903*   4979    4982    4988    4992    4996    5003    5008    5013    5018    5023    5030
                       5035    5040    5045
TYPES  = 104406         971     988    1840    2885    4176    4180    4904*   5056    5066
UNCMP    001164         845*    928*    951*    957     977*    983*    991*   1945    1966
UNITSV   001162         844*    927*    933     936     950*    957    2027    2134    3518
UNNUM    001160         843*    919*    929     952*    959     970     979*    984*    987    1007    1022    1072    1085
                       1174    1369    1372    1412    2400    2407    3534    3563    4924
WAITRY   025546        4932*
WC     = 000010         806*   1059    1115    1219    1223    1227    1248    1385    1389    2646    2649    2856    2859
                       3079    3082    3288    3291    3485    3488    3491    3547    3660    3939    3947    3950    3953
                       4095    4110    5014
WCE    = 040000         824*   4129
WCRC     001200         851*   4552*   4577    4590
WK15     001212         856*   4553*   4556*   4566
WORK     001214         857*   1323*   1324*   1325    1338*   1339*   1340    1346*   1347*   1348    1410*   1416*   1440*
                       1441*   1838*   1839    1906*   1920*   1951*   1956*   2364*   2371*   2561*   2582*   2586*   2587*
                       2596    2599*   2600*   2805    2882*   2884    3027    3527*   3528*   3858*   3879*   3883*   3884*
                       3898    3901*   3902*   4119*   4120*   4175    4179    4333*   4337*   4340    4441    4484*   4487*
                       4490*   4566*   4576*   4579*   4589*   4592*   4606*   4611    4613    4932*   4935*   4936
WORK0    001216         858*   4541*   4552    4555*   4568*   4571*   4582*   4585*   4595*   4598*   4606
WORK1    001220         859*   2375*   2406    4334*   4335    4338*
WORK2    001222         860*   5051*   5052*   5059*   5061*   5063*
WORK3    001224         861*   2782*   3004*   3217*   3414*   4247*   4251    4262*   4408    4417*   4421*   4429*
WORK4    001226         862*   5050*   5053*   5055
WORK5    001230         863*   4078*   4079*   5057*   5067*
WORK6    001232         864*   5058*   5060*   5062*   5064*   5065
XBIT   = 104432        2568    3865    4914*
XXOR     024056        4567    4581    4594    4611*
ZERONE   004476        1407*
$ENDAD   021200         710    4152*   4726
$END1    021210        4150    4154*
.      = 030210         708*    709*    711*    713*    716*    759*    770*    908     909*    914     931     932*    955
                        968     990     991*   1020    1024    1029    1034    1040    1046    1052    1058    1064    1083
                       1087    1092    1098    1104    1109    1114    1127    1131    1135    1143    1153    1163    1167
                       1172    1180    1184    1188    1199    1210    1218    1222    1226    1237    1247    1255    1259
                       1263    1274    1284    1292    1296    1303    1309    1316    1326    1333    1341    1349    1360
                       1364    1376    1384    1388    1396    1400    1422    1427    1447    1480    1626    1664    1671
                       1837    1859    1860*   1868    1869*   1949    1950*   1962    1969    1976    2032    2139    2173
                       2192    2224    2225*   2243    2278    2279*   2294    2325    2326*   2341    2384    2388    2393
                       2397    2413    2648    2651    2654    2858    2861    2883    3081    3084    3290    3487    3492
                       3493*   3546    3565    3615    3651    3665    3669    3691    3696    3700    3706    3710    3946
                       3949    3952    3955    4052    4114    4131    4132*   4148    4149*   4156    4161    4174    4196*
                       4341    4639    4644    4697    4712    4713*   4718    4719*   4721    4732    4733    4742    4762
                       4765    4770    4780*   4813    4817    4856    4870    4876    4887*   4977    4980    4981*   4986
                       4990    4991*   4994    5001    5002*   5006    5007*   5011    5012*   5016    5017*   5021    5022*
                       5028    5029*   5033    5034*   5038    5039*   5043    5044*   5049    5050*   5069    5070*   5074*
                       5075*
.CLKD1   022032        4279*   4915
.CLKD2   022126        4310*   4916
.CLKR1   023010        4439*   4917
.CLKR2   023240        4479*   4918
.CLROK   025510        4907    4923*
.DSCK    021604        4220*   4912
.GETSP   025630        4920    4957*
```

```
.HLT    024432              896    4706#
.HLTCT  024600             4715*   4732#   4975    4997    4999    5004    5009    5014    5019    5025*   5026    5031    5036
                           5041
.ILLUP  025142             4789    4816#
.KIT    024402             4668    4677    4683    4690#   4730
.MRCK   021540             4208#   4909
.MRCLK  021510             4202#   4910
.MROMD  025536             4908    4929#
.MRIND  021566             4217#   4913
.MRINT  021502             4198#   4911
.OVER   024406             4666    4691#
.POW_R  025014              894    4789#   4811
.POWUP  025060             4798    4801#
.PR     024770             4739*   4741*   4743*   4744*   4749    4753    4763*   4764    4767*   4769    4773    4780#
.PRF    024676             4751    4754    4759#
.PRL    024654             4752#   4768
.PTIT   024634             4740    4745#
.PUVEC  025150             4789*   4790*   4798*   4820#
.QQ.  = 000052             4888#   4901    4902#   4903#   4904#   4905#   4906#   4907#   4908#   4909#   4910#   4911#   4912#
                           4913#   4914#   4915#   4916#   4917#   4918#   4919#   4920#   4921#   4922#
.RBIT   022640             4408#   4919
.RDLIN  025272             4865#   4906
.RDOCT  025154             4826#   4905
.SAVR6  025146             4797*   4801    4819#
.SCOPE  024240             4663#   4901
.SPASS  025670             4921    4968#
.SVLAD  024362             4674    4679    4685#   4693
.SWOPT= 163400              692#    699     700     701     702     703     704     705     706     707     708    4146    4147
                           4149    4663    4706    4710    4723
.TBIT   021214             4156#
.TRAP   025416              898    4895#
.TRP    025434             4898    4899#
.TYPE   024102             4626#   4902
.TYPEB  024602             4739#
.TYPEO  024612             4741#   4903
.TYPES  024622             4743#   4904
.XBIT   021632             4234#   4914
```

```
END          1#     692#
NEWTST       1#     692#    1002     1066     1116     1137     1146     1175     1191     1203     1212     1228     1242     1249     1265
          1279     1286     1299     1305     1311     1318     1328     1335     1343     1351     1367     1379     1391     1403     1431
          1450     1628     1802     1890     1931     1984     2034     2141     2194     2246     2296     2343     2416     2656     2889
          3086     3297     3500     3569     3619     3671     3713     3958     4066
ODT11        1#
ODT11X       1#
POP          1#     756#    4673     4676     4774     4805     4850
PRINT        1#     756#     908      914      931      955      968      990     1837     1859     1868     1949     1962     1969     1976
          2173     2224     2278     2325     2413     2883     3492     3651     3665     4114     4131     4148     4161     4174     4341
          4712     4718     4813     4855     4875     4977     4980     4986     4990     4994     5001     5006     5011     5016     5021
          5028     5033     5038     5043     5049     5069
PUSH         1#     756#    4670     4745     4791     4828
SCOPE.       1#    1002#    1066#    1116#    1137#    1146#    1175#    1191#    1203#    1212#    1228#    1242#    1249#    1265#    1279#
          1286#    1299#    1305#    1311#    1318#    1328#    1335#    1343#    1351#    1367#    1379#    1391#    1403#    1431#    1450#
          1628#    1802#    1890#    1931#    1984#    2034#    2141#    2194#    2246#    2296#    2343#    2416#    2656#    2889#    3086#
          3297#    3500#    3569#    3619#    3671#    3713#    3958#    4066#
SET       4888#    4901     4902     4903     4904     4905     4906     4907     4908     4909     4910     4911     4912     4913     4914
          4915     4916     4917     4918     4919     4920     4921
TYPADR       1#
TYPBIT       1#     756#
TYPCHR       1#     756#
TYPDEC       1#     756#
TYPLIN       1#     756#
TYPOCS       1#     756#     969      987     1839     2884     4175     4179     5055     5065
TYPOCT       1#     756#    4716     4978     4981     4987     4991     4995     5002     5007     5012     5017     5022     5029     5034
          5039     5044
TYPTXT       1#     756#     907      913      931      955      968      990     1837     1859     1868     1949     1962     1969     1976
          2173     2224     2278     2325     2413     2883     3492     3651     3665     4113     4131     4160     4174     4341     4977
          4980     4985     4989     4993     5001     5006     5011     5016     5021     5028     5033     5038     5043     5049     5069
$CATCH       1#     692#     708
$CMTAG       1#     692#     759
$DONE        1#     692#    4141
$EQUAT       1#     692#     725
$HLT         1#     692#    4698
$KMMR        1#
$OCTAL       1#     692#    4733
$POWER       1#     692#    4781
$RAND2       1#
$RAND4       1#
$RDDEC       1#
$RDLIN       1#     692#    4858
$RDOCT       1#     692#    4821
$SCOPE       1#     692#    4656
$SET      4888#    4901     4902     4903     4904     4905     4906     4907     4908     4909     4910     4911     4912     4913     4914
          4915     4916     4917     4918     4919     4920     4921
$SETUP       1#     692#     893
$SMMR        1#
$SWDOC       1#     692#     696
$TRAP        1#     692#    4888
$TYPE        1#     692#    4616
$TYPEA       1#
$TYPED       1#
$UMMR        1#
```

```
. ABS.  030210     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 .DERSDC SEQ/SOL/CRF/NL:TOC=DERSDC.SML,DERSDC.SRC/DS:ERFZ
 RUN-TIME: 9 14 1 SECONDS
 RUN-TIME RATIO: 540/25=21.4
 CORE USED:  22K  (43 PAGES)
```